(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
30 August 2001 (30.08.2001)

PCT

(10) International Publication Number
WO 01/63472 A2

(51) International Patent Classification[7]: G06F 17/30

(21) International Application Number: PCT/IL01/00173

(22) International Filing Date: 22 February 2001 (22.02.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/184,803 24 February 2000 (24.02.2000) US

(71) Applicant *(for all designated States except US)*: **BMI-DAS.COM LTD.** [IL/IL]; Simtat Shai Agnon St. 8, 65200 Givat Shmuel (IL).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **TUR, Ziv** [IL/IL]; Simtat Shai Agnon St. 6, 65200 Givat Shmuel (IL). **BEN DAVID, Tzvi** [US/IL]; Menahem Begin Rd. 58, 97000 Petah Tikva (IL). **BILLER, Koby** [IL/IL]; Rupin St. 39, 76353 Rehovot (IL).

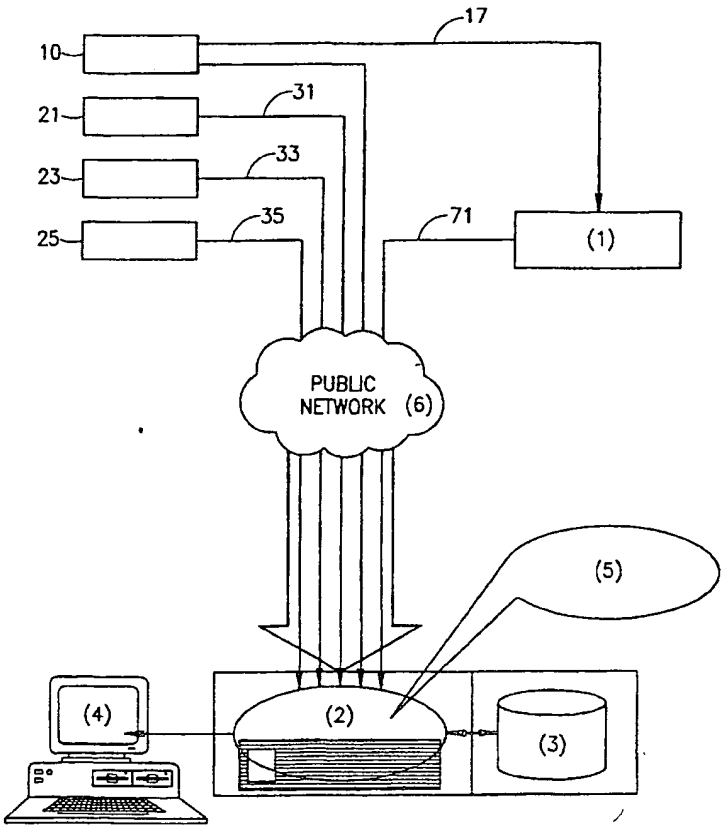
(74) Agent: **CHIRNOMAS, Mordechai**; Shiboleth Yisraeli Roberts Zisman & Co., Montefiore St. 46, 65201 Tel Aviv (IL).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: SYSTEM AND METHOD FOR SECURE, QUERY-DRIVEN, TARGETED ELECTRONIC SOLICITATION



(57) Abstract: A system and method for directing a blind solicitation to a pre-definable, anonymous potential customer client via the network. A supplier sends an offer to the query-aggregating server, communicatively connected to a network. The query-aggregating server receives a recognized request and sends a query to the client system through the network. The query comprises an offer from the supplier and a definable characteristic profile for identifying a potential customer .In response to the query the client system activates the personal agent, located in the client agent. The personal agent executes the query, scans the data in the personal database, located in the clients system, and determines the relevance of the offer to the customer.

WO 01/63472 A2

Published:

— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

SYSTEM AND METHOD FOR SECURE, QUERY-DRIVEN, TARGETED ELECTRONIC SOLICITATION

Field of the Invention

The present invention relates to the field of network navigation and browsing. More particularly, the present invention relates to a method and system for securely directing highly focused, blind solicitations from soliciting suppliers to a pre-definable, potential customers while maintaining the potential customer's anonymity.

Background of the Invention

The information network known as the World Wide Web (WWW) is arguably the most complete source of publicly accessible information available. Today, companies offer various subscription services accessible via the network.

For example, many people now do their baking, stock trading, shopping and so forth from the comfort of their own homes via Internet access. Companies from various industries, utilize the advantages of the network to expand the number of their potential customers.

The main problem many companies are facing is targeting and identifying those anonymous potential customers. With so many people of various interests and needs surfing the network, it's difficult for a company to be sure that the distributed information and data will indeed reach the desired target audience, and answer their specific needs, thus creating that important first link with future prospective clients.

The paradox of reaching as many people as possible while maintaining a good unique and personal relationship calls for a system which can recognize potential clients and make sure vital information is made available to them in such way that benefits both sides.

The situation of a fast, personal and diverse communication world created new customers. Those customers demand relevant and available data in real time. The data they receive from the suppliers must fit their economical and social conditions and ambitions.

Customers connected to the network are facing the same problem from the other side. Until recently the sheer volume of information available, strewn over various sites and sources could leave a would-be customer hopeless and confused. This situation creates a frustrated customer. Customers today receives mass of data that isn't relevant to them, meanwhile the data they need and that would interest them often does not reach them.

What is needed is a system that will ensure a customer that he will get all the data and information he needs, data that matches his personal conditions. Maintaining a productive relationship over the network today is first and foremost a matter of privacy and security. In today's business world where some of the main Internet services providers are banks and credit companies, matters of secrecy and security are of grave importance. From the very first steps, a customer is usually asked to provide sensitive private information that is essential to the relationship at hand. The client expects a fully secured system, assuring that none of his personal data will be made available to other parties.

More particularly, there is a need for a system which permits soliciting suppliers, i.e. companies which want to identify particular classes of customers and offer goods or services to those particular customers. Balanced against this need is the potential customer's desire to be identified for offers which are particularly well-suited to his needs or lifestyle without being bombarded by junk e-mail, cel phone calls, SMS text ads, etc. and to be identifiable to the soliciting suppliers without exposing any personal information to access by the soliciting supplier or any other third party.

Summary of the Invention

Thus the present invention has the following as objectives, although this following is not exhaustive:

It is one objective to permit soliciting suppliers to be able to identify with greater particularity those persons which it considers as potential customers for select products, services and offers.

It is a further objective of the present invention to be able to identify particular potential customers by being able to query the information in the databases on the personal computers of the potential customers.

It is a further objective of the present invention to permit polling or querying of databases on personal computers of potential customers without actually giving access to the personal database to the enquiring soliciting supplier.

It is a further objective of the present invention to permit polling or querying of databases on personal computers of potential customers without actually identifying oneself to the enquiring soliciting supplier.

It is a further objective of the present invention to permit potential customers to filter with greater accuracy the kind of unsolicited offers and solicitations they receive by permitting anonymous polling or querying of personal databases without without actually giving access to the personal database to the enquiring soliciting supplier.

It is still a further objective of the preset invention to provide a potential customer's client device with a personal agent which can execute queries originating from a soliciting supplier and interact with the local personal database on the client device and determine whether or not to bring the query to the attention of the potential customer.

It is yet a further objective of the present invention to provide personalized solicitation screening to potential customer's which can refer to the potential customer's personal databases with security being guaranteed to the potential customers by providing a system in which personal information is only inflowing, never outflowing, during the polling or querying process.

These objectives and others not mentioned hereinabove are accomplished by the methods and systems of the present invention in which a blind offer may be directed to a pre-definable anonymous potential client device. In one exemplary embodiment of the present invention, a supplier, for example third party desiring to present an offer to a potential customer's client, is registered with the server of a query-aggregating service provider and is provided with software for creating a query, supplied by the query-aggregating server company. The query comprises an offer from a supplier and a definable characteristic profile for identifying a potential customer. After the soliciting supplier creates the query he sends the query directly to the query aggregating server (which may also be referred to hereinbelow as the service provider server).

In an alternative embodiment of the present invention, the details of the offer are simply forwarded from the soliciting supplier to the query-aggregating service provider by an means including by facsimile, telephone, speech recognition, etc. the query is then formulated by the query-aggregating service provider into a downloadable query and saved in the query-aggregating and forwarding server. Please note that the gathering and forwarding function are independent and can be performed by independent, but communicatively linked, servers or by a single server performing both functions.

The customer's client device, communicatively connectable to the network, includes a personal agent of the preset invention, for requesting and receiving queries from the query aggregating server via any communications network, and a personal database accessible to the personal agent. The personal agent includes a program, the database management system (hereinafter "DMS") for executing a query, scanning data in the personal database and determining the relevance of the personal database contents to the query in question. When such relevance is established the personal agent notifies the potential client of the offer, while no identifying information about the client is returned or revealed to the query-aggregating server or to the supplier.

Brief Description Of The Drawings

The details of the present invention, both as to its structure and operation, can best be

understood by referring to the accompanying drawings, in which like reference numbers and designations refer to like elements.

FIG. 1a is a block diagram of a typical conventional Internet communication system where the database is located on the server;

FIG. 1b is a block diagram of a typical conventional Internet communication system where the database is located on the client;

FIG. 2a is a block diagram of one embodiment of an Internet communication system, according to the present invention;

FIG. 2b is a block diagram of one embodiment of an Internet communication system, according to the present invention;

FIG. 3a is a block diagram of an exemplary client system, shown in FIGS. 2a and 2b;

FIG. 3b is a block diagram of an exemplary client system, shown in FIGS. 2a and 2b;

FIG. 4a describes the types of elements, according to the present invention;

FIG. 4b describes one embodiment of the elements described in FIG 4a;

FIG. 5 is an exemplary Flowchart diagram;

FIG. 6 is a block diagram of a network constructed in accordance with the present invention;

Fig. 7 is an overview of a blind solicitation system, according to an embodiment of the present invention; and

Fig. 8 is an illustration of the client's device connected to the network, according to an embodiment of the present invention.

Detailed Description Of The Exemplary Embodiments

A typical known Internet communication system is shown in FIG.s 1a and 1b. With reference to FIG. 1a, a client computer system 102 communicates with server computer system 106 across the Internet 108. For simplicity, only one client and one server are shown, although many servers and clients are actually connected to the Internet. Client system 102 executes a browser application program 104 that allows a user of client system 102 to access objects, such as documents, graphics, programs, etc., that are stored on a server, such as server 106, through the Internet 108. The browser 104 displays graphics and/or text, which represents, identifies or describes objects, which may be accessed. The user selects an object 106 to be accessed, typically by clicking on the text or graphics representing the object. Each object is identified by an Internet address known as a uniform resource locator (URL). For example, URL 114 identifies object 116, which is stored on server 106. The URL specifies the location of an object on the Internet, including the server on which the object is located and the location of the object on that server. Browser 104 stores a URL identifying each object 116 on serverv 106 which is available and for which text or graphics may be displayed by the browser.

In response to the selection by the user of an object, such as object 116, which is identified by URL 114, browser 104 uses URL 114 to initiate an access 110 to object 116. Server 106 then initiates a transmission 111 of object 106 to client system 102. Each object includes an identifier of the type of the object. In FIG. 1, object 116 includes type identifier 112. Common object types include, for example, "Hyper Text Markup Language" (HTML) objects and "Graphical Interchange Format" (GIF) objects. The

object type identifier allows the browser to properly process and display the received object. The object type is not usually communicated to the user.

With respect to a database-related application, the server 106 connects with the database 120 and retrieves data. The Server 106 then initiates a transmission 111 of object 106 to client system 102, via the publicly accessible and vulnerable Internet. The transmission 111 includes data gained from database 120 after being analyzed by the server 106. Thus personal data, which was stored on database 120, has been exposed to the server 106 and then exposed on the Internet connection 108.

FIG. 1b differs from FIG. 1a by where the database is located. In FIG. 1b, it is shown that when the database 170 relates with client 152, client 152 initiates a transmission 160 to the server 156. Transmission 160 includes data extracted from the database 170. The data is then analyzed on the server 156 and transmitted back to the client 152 by the server 156. Again, in this scenario, personal data has been transferred via the unsecured Internet connection 160, has been analyzed on the server 156 and sent back through connection 162. Personal data has again been exposed on connection 160, then exposed to the server 156 and then exposed again on connection 162.

The present invention relates to a system and method whereby a query, which usualy will comprise an offer from a soliciting supplier, is programmed and delivered, via a query-aggregating service provider server, in what is an essentially one-way communication path, to be read only by those potential customers for whom the offer will be of particular interest. The query is constructed in a manner which may be authenticated and executed by a personal agent in the potential customer's client device. An analysis is then performed by the personal agent on the contents of the query and based upon comparison of the query parameters with the personal database of the client device. The personal agent uses the comparison analysis to determine whether in fact the offer is one which is likely to be of interest to the potential customer. If the decision is that the offer is of interest or relevant to the potential customer's

interests profile, then the personal agent brings the offer to the attention of the potential customer. Thus, without having to open many uninteresting, unsolicited junk solicitations, a potential customer can passively receive offers that were truly designed for him as a consumer and the soliciting supplier can know that only appropriate consumers, such as those with real means to accept the offer, will even be made aware of the special terms of the offer. Instead of mass mailings, mailings can be directed to preferred customers, even preferred customers of which the soliciting supplier was unaware.

The above process is carried out with reference to the customer's personal information available on his local database, according to the specific demands and requirements of the soliciting supplier and yet without permitting the soliciting supplier to either know the identity of the potential customer nor to have access to any of the extremely personal and sensitive information stored in the user's database. The following exemplary embodiments describe alternative ways in which the system and method of the present invention may be constructed and carried out. Athough reference may be made to certain specific types of networks, comunications devices, browser environments and software languages, it should be understood that these are being used by way of example only and are not necessarily limited to the precise examples given herein.

With reference to FIG. 2a, there is illustrated a block diagram of one leg of an embodiment of an Internet communication system for focusing offers or advertisements of goods or services to particular customers while still maintaining the anonymity of blind solicitation, in accordance with the present invention. A personal agent **240** has been previously installed and made a part of a user's (potential customer's) client resources in the client device **200**, i.e. everything on the client's side of the network **208** between client **202** and query-aggregating service provider server **206**. Personal agent **240** may be constructed according to one exemplary embodiment as described further hereinbelow in Appendix A and functions as the go-between for the client's resources **200** and query-aggregating server **206**. Personal agent **240** comprises a query processing bundle which includes the data management system **230** (usually

incorporating some functional portion or extension of a device's operating systems), through which personal databases **220** are built, managed and accessed, all on the client **202**. Personal agent **240** may further comprise data filters and object display handlers for dealing with the transfer of information between the DMS **230** and the client browser **204**. A user of client system **202** selects an object **241** identified by URL **214**. URL **214** identifies an object **241** on personal agent **240**, which includes a request **232** for additional data, i.e. the user initiates a query dowload request sequence from a server **206**. Personal agent **240** initiates request **232** of DMS **230**, which initiates transmission **222** of request **232**. Database **220** receives transmission **222** of request **232** and sends the results of the transmission to DMS **230** via transmission **221**. The request is analyzed by personal agent **240** using DMS **230** and is returned, tagged irrelevant (i.e., not of interest to the user, not a potential customer) or relevant (suitable for and of interest to the user or potental customer) and enriched with personal data, to the personal agent **240**, which sends it to the client **202**, if relevant.

The objects **241** on personal agent **240** are kept up to date by the server **206** using periodical push technology as shown later in FIG. 6. The object **241** has a clear definition in it of what result is relevant and what result isn't relevant. Using the invention, no personal data has left the client personal resources. The only data passed over the Internet is a request for query download and the handshake processes. Substantive data of a fairly general nature only, i.e. the query itself, is passed only in the direction of user's device **200**.

With reference to FIG. 2b, an alternative exemplary embodiment has objects **266** which are stored on server **256**. A user of client system **252** selects an object identified by URL **264**. URL **264** identifies an object **266** on query aggregating server **256**, which object **266** includes a request **263** for additional data as well as an offer forwarded to server **256** by a supplier (not shown). Object 266 is passed over a network, such as the Internet, to browser **254** which opens the object **266** and sends the request **263** to personal agent **290**. Personal agent **290** initiates an access **282** of DMS **280**, which initiates a transmission **272** of request **263**. The request is analyzed by DMS **280** with reference to personal databases **270** and returns **281** tagged either irrelevant or

relevant to personal agent **290**. If tagged relevant, the request **263** is first enriched with the personal data culled from the personal databases **270** before being returned to personal agent **290**. Personal agent **290** sends enriched relevant requests **263** to the browser **254** to be displayed on client **262**. Irrelevant tagged requests **263** are simply deleted, trashed or otherwise disposed of, although they might, for example, be saved for casual browsing. Using the invention, no personal data has left the client personal resources and yet the user has been made aware of an offer of particular relevance to his interests without the supplier even being actually aware of his identity. Additionally, the supplier has been able to send an offer which it can be confident will only be displayed for users for whom such an offer is really of interest. The only data passed over the Internet is general information contained in object **266**, i.e., the offer and the criterion by which the query can determine relevance of the offer to the recipients thereof. The criterion comprise the request **263** which will be made to the personal databases **270** and analyzed for relevance or irrelevance.

An altenative exemplary embodiment of the system and method of the present invention are described hereinbelow where the client system is shown based on a personal computer ("PC") as illustrated in FIG. 3a . The arrangement described hereinbelow is well-suited, for example, where a bank server sends information directly to a user and then later uses the query aggregating server to send focused queries to its customers according to the present invention. The embodiment is divided into 4 stages for ease of understanding:

Setup Phase: In this stage, the system, which includes a service provider server **305**, sends data **307** to personal agent **340** upon getting a request from the personal agent **340**. Personal agent **340** sends the users personal data **307** to the DMS **330** through transaction **332** for saving in a personal database **320** via transaction **322** and from which the DMS **330** will be able to access the information in the future in response to queries. An example of this might be the stage when all of the user's bank account details from a particular banking services provider are first downloaded and saved locally.

General Update Phase: Server **306** sends general data, for example interest rates, exhange rates, general information regarding accounts, etc. and interface objects

(HTML) to personal agent **340** through transaction **312**. The general data is then transmitted through **332** to the DMS **330** and then stored on database **320** through transaction **322**. This type of general data might be useful for updating objects used for displaying results of relevant queries.

Query Execution Phase: Client **302** accesses object **341**, which is identified by URL **314**. Object **341** is stored on personal agent **340**. When object **341** is activated, it sends a data request **332** to the DMS **330**. The DMS **330** imports the saved users personal data acquired from the setup phase and from previous general data updates, as well as the current query from personal database **320** and sends it back to the personal agent **340** through transaction **331**. The personal agent **340** then sends the enriched object **341** to client **302** through transaction **311**. An enriched object **341** is the offer which will be displayed to the browser when a query has been tagged relevant, however the offer now may have incorporated therein personal data taken from the personal database **320** which is specifically relevant to the substance of the offer.

Object Enrichment: As a hypothetical, a bank A may wish to find potential customers who, according to their own personal data have a savings account with another bank having a minimum balance of $5,000 and are earning interest at a rate of 6.5% or lower, for the purpose of offering those persons an account which will bear interest at 7%. The details of the sought customer will be contained in a search profile table in the query and the details of the offer will be in an offer HTML object in the query. When the DMS **330** analyzes personal database **320** and finds that the user has a savings account #123456789 at bank B which has a balance of $10,000 and earns interest at a rate of 5.5%, the query is tagged as relevant, the object is enriched with the details about the user's current account situation, and the enriched offer is displayed, for example as follows:

"Bank A is pleased to advise you that the $10,000 which you have on deposit in Account #123456789 at Bank B earning only 5.5% interest is eligible to be deposited into one of our high-interest accounts bearing interest at 7%. If you would be interested in more details, please either press on the button to send us an automatic e-mail or contact Joe A. Banker at the following phone, fax, etc. "

One way of constructing the above mechanism is shown in appendix A, Module 2 where it can be seen that object **341** is an HTML object. In the HTML code of object **341** there is a request for data. The HTML activates the Active X control shown in module 1 of Appendix A to execute a data request from the database and then sends the enriched object to the client (transaction **311**).

Periodic Query Re-execution Phase: In this phase, If-Then queries, which have been saved on the client 302 are re-executed periodically by personal agent **340** and DMS **330** re-analyzes the clients' database **320** for the purpose of determining whether a previously irrelevant query is now relevant due to changing personal circumstances in the user's life as evidenced by changes in the personal databases **320**. For example, the Bank A query would not be displayed if no accounts exceed $5,000, however several months, there may be accounts which now satisfy the minimum balance requirement of the query and should now be of interest to the user. Personal agent **340** initiates a set of analyses to be made by the DMS **330**. The DMS **330** runs the queries using the data stored in the Setup and General Update Phases **307 and 312** on database **320** by transaction **322**. The analyses are then sent to the personal agent **340** through transaction **331**. The analyses are enriched with the clients' personal data and tagged relevant/irrelevant this is done by the DMS using the database. The personal agent **340** then sends the analysis tagged relevant to the client **302** using transaction **311**. The analyses are displayed on the client's browser **304**. This phase is typically run on many objects which have been stored over a period of time. Although the agent may have processed many analyses, the client receives results in the form of the offers only of those queries whose analysis indicated that they are relevant to the specific interests of the potential customer.

With reference to FIG. 3b, there is illustrated an alternative exemplary embodiment in which the client device is a personal data assistant (PDA) or any other wireless communications-enabled device such as a wireless application protocol (WAP)-based client system, i.e. an Internet-enabled cellular phone is shown in FIG. 3b:

Phase 1 (Setup): The system in the figure includes a service provider server **355**, which sends data **357** to personal agent **390** at the request of personal agent **390** and only after the appropriate security clearance procedure has been performed by personal agent **390**. Personal agent **390** sends the user's personal data to the DMS **380** through transaction **382**. The DMS **380** saves the data on database **370** through transaction **372**.

Phase 2 (Query Download and Execution): Client **352** accesses object **366**, which is identified by URL **364**. Object **366** is stored on personal agent **390**. When object **366** is operated, it sends a data request **382** to the DMS **380**. The DMS **380** imports the data from database **370** (enrichment) and sends it back to the personal agent **390** through transaction **381**. The personal agent **390** then sends the enriched object to client **352** through transaction **361**.

Phase 3 (Periodic Query Re-execution) : Personal agent **390** periodically analyzes the clients' database **370**. Personal agent **390** initiates a request **362** to the browser **354** which accesses object **366**. Object **366** is identified by URL **364** and is stored on server **356**. When object **366** is operated it initiates a set of analyses to be made on the DMS **380**. The DMS **380** runs the queries using the data stored on database **370** with transactions **382** and **381**. The analyses are then sent to the personal agent **390** through transaction **381**. The analyses are enriched with the clients' personal data and tagged relevant/irrelevant all of which is done by the DMS **380** using the database **370**. The personal agent **390** then sends offers of those analyses which were tagged relevant to the client **352** using transaction **362**. The offers which survive the analyses and have been tagged as relevant are displayed on the client's browser **354**. This phase is run on many objects. Although the agent **390** processes many analyses, the user may have displayed only the few offers which the analyses showed were of relevance to him.

In FIG. 4a we see an illustration of the client system. The client system is divided into the following four elements:

Examples of browser applications **411** which may be used in the present invention include an Internet browser suitable for navigating the Internet from a desktop PC or network terminal, such as Microsoft Corporation's Internet Explorer®, Netscape Corp.'s Navigator® or it may be a wireless browser designed for display using the Wireless Application Protocol ("WAP") **412** or some other application useful on a protocol designed for use in the wireless device environment, e.g. cellular phone browser application **413** or a wireless personal digital assistant, or in fact for any other form of network-enabled application such as WebTV, in which the client has no local resources at his command, i.e. the player is temporarily downloaded along with the client.

A personal agent **421** may be an ActiveX control (OCX) module implemented from within the browser application **411** as a browser application plug-in module **422**, and implemented in the browser application itself. Alternatively, the personal agent **421** may be a Java applet **423**, implemented from within browser application **411**. Finally, personal agent **421** may be constructed as a stand alone application **424** executable directly in the operating system of choice for the intended recipient.

Alternative exemplary embodiments may use a database management system **431** (hereinafter "DMS") which is based on protocols such as an ODBC Protocol **432**, JDBC Protocol **433**, or ADO / DAO Protocol **434**. For using specific database commands which are not implemented by the particular protocols, it is possible to construct a database stored in text files and implement code written in Visual Basic to retrieve data from the text files as demonstrated in Appendix A, module 7.

A database 441 useful for the present invention can be practically any form of database, such as Microsoft Corporation's Access® and Fox Pro®, text files, SQL Server®, Oracle®, Palm OS-compliant, Windows CE compliant, etc.

With reference to Fig. 4b, there is illustrated an alternative exemplary embodiment of the client system of the present invention.

In the instant exemplary embodiment, the four basic elements, browser, personal agent, DMS and database, may be located and implemented on the client's system, e.g. a PC or other network browser-enabled device.

The flowchart diagram shown in FIG. 5 illustrates an example of the processing of a query.

A request for using the personal agent may either initiated by the user manually or automically according to a periodic interval estabished by the user or the agent itself, for example everytime the browser is activated and at ten minute intervals:

1) User request: the user opens his browser and elects to look at his financial status, the agent is automatically activated. Automatic initiation may include that the agent identifies that the user is connected to the Internet and proceeds or the agent calculates that some specified time interval since the previous update has passed and, even in the absence of an Internet connection, establishes new Internet connection and connects by itself.
2) Personal agent requests an update of the objects from the query aggregating server of the personal agent service provider. The personal agent downloads the updated list of object files from the query aggregating server, and compares the updated list with a local (to the client) list, which is stored on the personal agent's system. The personal agent then downloads the updated or new object files and deletes the unnecessary files.
3) The personal agent activates an object. For example, the object may be an HTML file which the personal agent reads and identifies as being requests for data, in one exemplary embodiment accompanied by a soliciting supplier's offer.
4) The activated object requests data from the DMS, using the DMS commands.
5) The request is then processed by the DMS using the database. The DMS returns the request tagged relevant or irrelevant and enriched with personal data from the database. For example, if a request was sent by a bank to check if the user is in overdraft and if so in what different accounts he has a positive balance, for the purpose of offering a new overdraft account feature or automatic account balance pooling service. The DMS checks in the database related to the banking accounts

and, if there is an overdraft, the request is returned tagged relevant. Then the DMS enriches the request with personal data, in this case other accounts where the user has a positive balance. For an account related to travel, the DMS would check the travel-related database or databases.

6) If the request was tagged irrelevant by the management system, the process ends. If the request was tagged relevant, it is passed to the client.

7) The client receives a data-enriched object. This object is then made available for the clients interface e.g. browser. The object can be shown to the client in different ways: desktop browser, WAP cellular phone browser, wired or wireless PDA, e-mail, etc.

A further alternative exemplary embodiment of the personal agent is illustrated in FIG 6, in which the query aggregating server provides the additional service of gathering general data and delivering it to personal agent equipped clients using the personal agent as a filtering mechanism. For example, the query aggregating server can be gathering news and stock index information on an ongoing basis from various news suppliers and Internet information portals. When the client signs into the system, the personal agent will be given a download of current events and stock information queries which will display only the particular news events stories and stock market information relevant according to the analysis of the query. As follows:

Step 1: The query aggregating service provider server **601**, gathers general data from Internet based general data providers **610**, the data is then stored on query aggregating service provider server **601**. The query aggregating service provider server **601** is fed with updated objects, by the agent services provider company **610**. The query aggregating service provider server **601** now holds general data and updated objects due to this process, which is repeated routinely.

Step 2: The agent **606** is activated, automatically or by the clients' request. On activation, the agent **606** connects to query aggregating service provider server **601** and updates its database with updated general data and updated objects stored thereon. Then the agent **601** connects with personal service providers **602** of the particular user that the agent **601** is serving. Examples of personal service providers are personal e-banking Websites or a travel agent Website. Of course, the service provider need not be contactable via the Internet as othyer forms of networks are specifically

contemplated to be utilized with the present invention. After gathering all the data, it is transferred to the agents database **607**.

Step 3: The objects in the personal agent **606** are activated, handled by the DMS (not shown but part of personal agent **606**) and returned from the personal database **607** tagged irrelevant or relevant and enriched with personal data. The objects can optionally be activated (displayed) by the user or automatically by the personal agent **606**. If relevant, the personal agent sends the enriched object for display on the client's preferred interface, for instance, browser **620**, cellular phone, PDA or WAP device **621** or e-mail **622**.

Fig. 7 is an overview of a blind solicitation system, according to an embodiment of the present invention. In step 1, databases and objects on the client device **2** are updated. Supplier's servers **10, 21, 23,** and **25** connected to the Internet **6** (or other network), send updated objects data and personal information **11, 31, 33,** and **35** to the client's personal agent **5**. The network **6** could be a cellular phone network, local area network ("LAN") or a wide area network ("WAN") such as the World Wide Web (WWW) or any other part of the Internet. The present invention isn't limited to any specific kind of network. The personal agent **5**, whether activated automatically or by the client's request, receives personal data and sends it to the personal databases **3**, where all personal data is securely kept. For example, server **21** may be bank server, wherein interactive on-line banking and account managing may be performed, sending personal data **31** which contains information about the client's bank account.

In order to achieve a high level of security the user shouldn't be able to transmit his data, which is stored in the client's personal database **3**, to another user. Techniques such as double scrambling, and security handshakes and passwords are used to secure the data.

In the next stage, a soliciting supplier **10**, or any third party that wishes to send a specialized offer directed at a certain kind of pre-definable customer, sends a request to make a query **17** to a query aggregating server **1** for targeting the pre-defined potential customer. In response, the query aggregating server **1** provides the soliciting supplier **10** with a software query model builder, which the soliciting supplier **10** uses to creates

his own query. Query **17** includes the offer to be displayed if the recipient, in this case the user of client **2**, is found to be a match, as well as a table defining the characteristics for which the DMS of personal agent **5** must look on pesonal databases **3** in order to identify a match and tag the query **17** relevant for display to the user/potential customer. The query **17** thus generated is packaged and sent to the query aggregating server **1**, where it is held for downloading by users either for a set time period or until replaced or cancelled by either soliciting supplier **10** or by query aggregating service provider **1**. Thus each query may be provided by the soliciting supplier **10** that sent it, with a different and unique query that has the ability to identify the most suitable client to his request.

Finally, the personal agent **5** connects to the query-aggregating server **1**. In a previous installation process, the user's client device **4** downloads the personal agent **5** that enables client **2** to receive queries and offers, and select and decide from what suppliers and vendors he wants to receive service or about which subjects he would like to receive offers. After connecting and passing through a security protocol for the purpose of establishing identities and authorizations, the user's personal agent 5 receives a query 71 (although in reality a personal agent will receive and process as many previously unknown queries as there are) comprising an offer from a previously selected soliciting supplier **10**. The personal agent **5** determines the relevance of the offer by scanning the personal database **3**. The scanning is conducted by seeking for matches to the definable characteristic profile that was made part of the query by the soliciting supplier **10**.

If matches are found, the answer to the query is "relevant" and the offer is displayed to the user **4**. The personal agent **5** may be activated automatically, every time the clients device is operated, or according to time periods the client selects, for example every 12 or 24 hours.

Security and anonymity for the user are provided by the system of the present invention since the sensitive information in the form of discriminating queries always flows in a uni-directional way from the query aggregating server 1 and soliciting supplier's servers to the client device **2** and never does a soliciting supplier know who actually was seletced to see the offer. Through the whole process there is definite

separation between the query aggregating server and the personal database to assure that the data used to decide if the offer is relevant is never exposed through the network.

Fig. 8 is an illustration of the client's device connected to a network, according to an embodiment of the present invention. A query is created by a supplier **33**, using special software supplied by the query aggregating service provider, and transmitted by communications process **103** to query aggregating server **91**. The query **61** could be updated as described hereinabove with respect to a General Update Phase according to the supplier's **33** needs, by receiving an updated query **62** through line **102** directly from the query aggregating server via the network. The update of the query could be made by the customer **33** and later be sent to the query aggregating server, or created by the query aggregating server according to the customers requests. Personal database **70**, located in the personal agent, stores updated data **71,72,73** sent by servers **31, 32,33**, possibly in objects which came packaged as queries. The objects might thus be used to update supplier templates and the like for helping in displaying relevant data or offers sent by the supplier using the invention. The personal data **71, 72, and 73** are displayed on the user's screen **40**, using a local data aggregation system **51**. The personal data could be summarized and be displayed to the client **41**.

In addition to the personal data, the personal database **70** is shown containing a query and an offer **61**, which were pulled automatically or by the user's request, from the query-aggregating server **91** through communication process **104**.

The personal agent **12** contains a user preference checklist **45** which enables the user to decide from a list of soliciting suppliers the query aggregating company is serving or a list of subjects, from which suppliers or on what subjects he wishes to receive offers. A data filter **52**, working as part of the DMS portion of personal agent **12**, receives the personal data **71,72** and **73** from the personal database portions of personal agent **12**, and executes queries **61**, and **62**, by scanning data **71, 72,** and **73**. In the end of the scanning process the selectable offers that were found relevant, and that matches the client characteristics, would be displayed to the client via a personal offer display **42.**

For example server **33** can be a travel agency looking for new potential clients and offering them a ski vacation. The travel agency sends a query "can you find the word ski" and an offer for a ski vacation in France **103** to the query aggregating server **91**. The personal agent **12** downloads the query containing the offer from the query aggregating server **91** and scans the personal databases **71**, **72** and **73** looking for the word "ski". If the answer to the query is "relevant" then the travel agency offer is displayed to the client.

In one exemplary embodiment of the present invention, the user can choose to respond to the offers by sending e-mail or a fax **44** or calling directly to the supplier. In another aspect of the invention, the client can activate an offer responding machine **101**. The offer responding machine **101** enables the client to choose from a list of suppliers, from whom the client received offers, and to whom the user wants to send a response. The machine **101** informs the user that he received offers, optionally showing him a list of suppliers that sent him offers, and asks the user to indicate 1) to which suppliers he wants to send a message 2) what is the message he wants to send.

In an alternate exemplary embodiment, the soliciting supplier could have queries integrated within the HTML of Website, whereby, when a personal agent-equipped client browses on the Website, the queries can be downloaded and executed locally by the client, thereby providing an enriched experience to the client. In such a way, a soliciting supplier can still deliver well-focused offers, enriched with the personal information from the personal database, and the user/potential customer can rest assured that the information in the personal database was not available to any third party, or even to the soliciting supplier.

It should be understood that exemplary embodiments described hereinabove are merely given by way of non-limiting example. It is understood and anticipated that modifications and variations on the above examples may be made by one of skill in the art without departing from the spirit and scope of the invention as that invention is hereinafter claimed.

Appendix A

Code Examples of invention

Convention used: X {Y} = X is an example of Y

Module 1 [Part of Personal Agent] - Example of agent embodiment as Active X Control (OCX)
The following code when compiled as an Active X control (OCX) which is used in an HTML page shown in Internet Explorer [Browser] executes data request 332 to database management system using ADO [DMS].

Open Connection - Opens an ADO connection to DB.MDB [Database (320) in figure 3a].
CloseConnection - Closes an ADO connection to DB.MDB [Database (320) in figure 3a].
AskQuery - [Data Request 332]
ReturnValue - Returns data from the database, this data can be tested in the code of the HTML page and if relevant can be enriched.

```
Dim connConnectionToDB As ADODB.Connection
Dim rs As ADODB.Recordset
Private Type udR
ecord
  row() As String
End Type
Dim arrResult() As udRecord

Public Sub OpenConnection()
  Dim strDB_Path As String

'  strDB_Path = App.Path
  strDB_Path = GetSetting("BMidas", "Init", "AppPath", "c:\sarit\modules\")
  If Right$(strDB_Path, 1) <> "\" Then strDB_Path = strDB_Path & "\"
  strDB_Path = strDB_Path & "DB.mdb"

'  Open a connection.
  Set connConnectionToDB = New ADODB.Connection
  connConnectionToDB.ConnectionString = _
"     Provider=Microsoft.Jet.OLEDB.3.51;" & _
"     Data Source=" & strDB_Path & ";" & _
"     Persist Security Info=False"
  connConnectionToDB.Open
End Sub
Public Sub CloseConnection()
  connConnectionToDB.Close
End Sub
Public Function AskQuery(query As String, numOfFields As Integer)
  Dim intLoopCounter
  Dim intRowsCounter
  Dim size As Integer
  Dim temp() As String
  Set rs = New ADODB.Recordset
  ReDim temp(0)
  ReDim arrResult(0)
  intLoopCounter = 0
  intRowsCounter = 0
  size = numOfFields - 1
  rs.ActiveConnection = connConnectionToDB
  rs.Open (query)
```

```
    If rs.EOF Then
      AskQuery = Null
    Else
      intCount = 0
      rs.MoveFirst
      Do Until (rs.EOF)
        For intLoopCounter = 0 To size
          temp(intLoopCounter) = rs.Fields(intLoopCounter)
          ReDim Preserve temp(UBound(temp) + 1)
        Next intLoopCounter
        arrResult(intRowsCounter).row = temp
        ReDim Preserve arrResult(UBound(arrResult) + 1)
        rs.MoveNext
        intRowsCounter = intRowsCounter + 1
      Loop
      GetHisahon = " "
    End If
    rs.Close
End Function
Public Function ReturnValue(ByVal rowNumber As Integer, ByVal fieldNumber As
Integer) As String
  ReturnValue = arrResult(rowNumber).row(fieldNumber)
End Function
Public Function GetSize() As String
  GetSize = UBound(arrResult)
End Function
```

Module 2 [Part of personal agent]- Example of HTML page that uses the Active X control shown on module 1 to execute a data request from database and then sends enriched object to client [transaction 311 in Fig 3a]

A. Data.html requests data from the database through OCX

```
<HTML>
<HEAD>
<META NAME="GENERATOR" Content="Microsoft FrontPage 4.0">
<TITLE></TITLE>
</HEAD>
<BODY>
<!--  If any of the controls on this page require licensing, you must
   create a license package file. Run LPK_TOOL.EXE to create the
   required LPK file. LPK_TOOL.EXE can be found on the ActiveX SDK,
   http://www.microsoft.com/intdev/sdk/sdk.htm. If you have the Visual
   Basic 6.0 CD, it can also be found in the \Tools\LPK_TOOL directory.

   The following is an example of the Object tag:
-->

<OBJECT ID="test"
CLASSID="CLSID:0B770616-A8AB-11D3-A324-0080AD7DBF90"
CODEBASE="data/Test.CAB#version=1,0,0,0" width="495" height="34">
<param name="_ExtentX" value="13097">
<param name="_ExtentY" value="900">
</OBJECT>


<SCRIPT LANGUAGE="VBScript">
<!--

   test.OpenConnection

   'get itra
   dim choose
   choose =2
```

```
        'a =test.AskQuery("SELECT Max(TnuotBank.Date) AS MaxOfDate,
Last(TnuotBank.Itra) AS LastOfItra, TnuotBank.HeshbonID, HeshBank.HeshbonName
FROM TnuotBank LEFT JOIN HeshBank ON TnuotBank.HeshbonID = HeshBank.HeshbonID
GROUP BY TnuotBank.HeshbonID, HeshBank.HeshbonName HAVING
(((Last(TnuotBank.Itra))<>0)) ORDER BY Max(TnuotBank.Date) DESC ", 4)
   'a =test.AskQuery("SELECT Max(TnuotBank.Date) AS MaxOfDate,
Last(TnuotBank.Itra) AS LastOfItra, TnuotBank.HeshbonID, HeshBank.HeshbonName
FROM TnuotBank LEFT JOIN HeshBank ON TnuotBank.HeshbonID = HeshBank.HeshbonID
GROUP BY TnuotBank.HeshbonID, HeshBank.HeshbonName  ORDER BY
Max(TnuotBank.Date) DESC ", 4)
   a =test.AskQuery("SELECT Max(TnuotBank.Date) AS MaxOfDate,
Last(TnuotBank.Itra) AS LastOfItra, TnuotBank.HeshbonID, HeshBank.HeshbonName
FROM HeshMaxDate INNER JOIN (TnuotBank LEFT JOIN HeshBank ON
TnuotBank.HeshbonID = HeshBank.HeshbonID) ON (TnuotBank.HeshbonID =
HeshMaxDate.HeshbonID) AND (HeshMaxDate.MaxOfDate = TnuotBank.Date) GROUP BY
TnuotBank.HeshbonID, HeshBank.HeshbonName ORDER BY Max(TnuotBank.Date) DESC
", 4)
        dim imax
   sizeItra=test.GetSize
        imax =sizeItra -1
   dim it(5,4),i,j
    for i=0 to imax
   for j=1 to 4
      Select Case (j)
         Case 1
          it(i,j) = test.ReturnValue((i),0)
         Case 2
         it(i,j) = test.ReturnValue((i),1)
         Case 3
         it(i,j) = test.ReturnValue((i),2)
         Case 4
         it(i,j) = test.ReturnValue((i),3)
         End Select
   next
   next

    'get tnua and value of tnua
   a= test.AskQuery("SELECT Itra , Value , Makor , Asmachta , Date, HeshbonID
FROM TnuotBank ORDER BY Date DESC", 6)
   dim tnua(200,6),p(2,7)
   dim sizeTnua
   sizeTnua=test.GetSize
        imax =sizeTnua -1
    for i=0 to imax
   for j=1 to 6
      Select Case (j)
         Case 1
         tnua(i,j) = test.ReturnValue((i),0)
         Case 2
         tnua(i,j) = test.ReturnValue((i),1)
         Case 3
         tnua(i,j) = test.ReturnValue((i),2)
         Case 4
            tnua(i,j) = test.ReturnValue((i),3)
         Case 5
            tnua(i,j) = test.ReturnValue((i),4)
      Case 6
         tnua(i,j) = test.ReturnValue((i),5)
         End Select
  next
   next

       'get tnua according to account number:
```

```
   a= test.AskQuery("SELECT HeshbonID  FROM TnuotBank Group By HeshbonID" ,
1)

        dim sizeHeshbon        'number of accounts
   dim tnuotall(10,200,5)      'array contain all tnuot
               'i -heshbon id
               'j -num of record
               'k - num of field
   dim tnuot(10)                    'array will hold heshbon ID's
   dim tnuotGodel(10)

   sizeHeshbon =test.GetSize

        for m=0 to sizeHeshbon-1
         tnuot(m)=test.ReturnValue((m),0)

   next
        for i=0 to sizeHeshbon -1
      x=tnuot(i)

      a= test.AskQuery("SELECT Itra , Value , Makor , Asmachta , Date FROM
TnuotBank WHERE HeshbonID='" & x &  "' ORDER BY Date DESC ", 5)
      imax=test.GetSize -1
               tnuotGodel(i)=imax +1
      for j=0 to imax
            for k=1 to 5
            Select Case (k)
                  Case 1
               tnuotall(i,j,k) = test.ReturnValue((j),0)
                 Case 2
               tnuotall(i,j,k)= test.ReturnValue((j),1)
                 Case 3
               tnuotall(i,j,k)= test.ReturnValue((j),2)
                 Case 4
                  tnuotall(i,j,k)= test.ReturnValue((j),3)
                 Case 5
                  tnuotall(i,j,k)= test.ReturnValue((j),4)
             End Select

        next
           next
  next

   'get visa and value of visa
  ' a = test.AskQuery("SELECT BDate , Esek , BSum , Asmachta ,HSum ,
Month([Hdate]) AS [month], Month([bdate]) AS monthB From visa Where (
Month([Hdate])= 10 And CardID = 601 )" , 7)
   'dim visa_601_10(11,5)
  'imax=test.GetSize -1
   'for i=0 to imax
  'for j=1 to 5
  '  Select Case (j)
      '    Case 1
  '     visa_601_10(i,j)=test.ReturnValue((i),0)
  '  Case 2
  '     visa_601_10(i,j) = test.ReturnValue((i),1)
      '    Case 3
  '     visa_601_10(i,j) = test.ReturnValue((i), 2)
      '    Case 4
  '        visa_601_10(i,j) = test.ReturnValue((i), 3)
      '    Case 5
  '        visa_601_10(i,j) = test.ReturnValue((i), 4)
```

```
'    End Select
'next
 'next
'get visa and value of visa
    a = test.AskQuery("SELECT BDate , Esek , BSum , Asmachta ,HSum  From visa
ORDER BY  BDate DESC" , 5)
    dim visa(200,5)
    dim sizeVisa
   sizeVisa =test.GetSize
   imax=sizeVisa -1
    for i=0 to imax
   for j=1 to 5
      Select Case (j)
         Case 1
         visa(i,j)=test.ReturnValue((i),0)
      Case 2
         visa(i,j) = test.ReturnValue((i),1)
         Case 3
         visa(i,j) = test.ReturnValue((i), 2)
         Case 4
            visa(i,j) = test.ReturnValue((i), 3)
         Case 5
            visa(i,j) = test.ReturnValue((i), 4)
         End Select
  next
   next

    'get visa and value of visa
    ' a = test.AskQuery("SELECT BDate , Esek , BSum , Asmachta ,HSum ,
Month([Hdate]) AS [month], Month([bdate]) AS monthB From visa Where (
Month([Hdate])= 11 And CardID = 601 )" , 7)
    'dim visa_601_11(11,5)
  'imax=test.GetSize -1
   'for i=0 to imax
  'for j=1 to 5
  '  Select Case (j)
       '   Case 1
  '     visa_601_11(i,j)=test.ReturnValue((i), 0)
  '  Case 2
  '     visa_601_11(i,j) = test.ReturnValue((i), 1)
       '   Case 3
  '     visa_601_11(i,j) = test.ReturnValue((i), 2)
   '        Case 4
  '        visa_601_11(i,j) = test.ReturnValue((i), 3)
   '        Case 5
  '     visa_601_11(i,j) = test.ReturnValue((i), 4)
   '    End Select
  'next
   'next

    'get visa and value of visa
'    a = test.AskQuery("SELECT BDate , Esek , BSum , Asmachta ,HSum ,
Month([Hdate]) AS [month], Month([bdate]) AS monthB From visa Where (
Month([Hdate])= 10 And CardID = 3047 )" , 7)
 '   dim visa_3047_10(11,5)
  '  imax=test.GetSize -1
   ' for i=0 to imax
'  for j=1 to 5
'     Select Case (j)
'           Case 1
'        visa_3047_10(i,j)=test.ReturnValue((i),0)
'     Case 2
'        visa_3047_10(i,j) = test.ReturnValue((i), 1)
```

```
'          Case 3
'        visa_3047_10(i,j) = test.ReturnValue((i), 2)
'           Case 4
'           visa_3047_10(i,j) = test.ReturnValue((i), 3)
'           Case 5
'           visa_3047_10(i,j) = test.ReturnValue((i), 4)
'      End Select
'  next
'    next

    'get visa and value of visa
'    a = test.AskQuery("SELECT BDate , Esek , BSum , Asmachta ,HSum ,
Month([Hdate]) AS [month], Month([bdate]) AS monthB From visa Where (
Month([Hdate])= 11 And CardID = 3047 )" , 7)
'    dim visa_3047_11(11,5)
'  imax=test.GetSize -1
'    for i=0 to imax
'  for j=1 to 5
'     Select Case (j)
'           Case 1
'        visa_3047_11(i,j)=test.ReturnValue((i), 0)
'     Case 2
'        visa_3047_11(i,j) = test.ReturnValue((i), 1)
'          Case 3
'        visa_3047_11(i,j) = test.ReturnValue((i), 2)
'          Case 4
'        visa_3047_11(i,j) = test.ReturnValue((i), 3)
'          Case 5
'             visa_3047_11(i,j) = test.ReturnValue((i),4)
'     End Select
'  next
'    next

   'get pik and value of pik
    'a = test.AskQuery("SELECT Osum , Nsum , HeshbonID , Update , Name, SDate
, Ribit FROM  Pikdonot ", 7)
     a = test.AskQuery("SELECT Pikdonot.Osum , Pikdonot.Nsum ,
Pikdonot.HeshbonID , Pikdonot.Update , Pikdonot.Name, Pikdonot.SDate ,
Pikdonot.Ribit FROM Pikdonot INNER JOIN PikadonMax ON Pikdonot.Update =
PikadonMax.Max]nnUpdate" , 7)

    dim pik(100,7)
   imax=test.GetSize -1
    sizePik = test.GetSize
    for i=0 to imax
   for j=1 to 7
      Select Case (j)
         Case 1
         pik(i,j)=test.ReturnValue((i), 0)
      Case 2
         pik(i,j) = test.ReturnValue((i), 1)
           Case 3
         pik(i,j) = test.ReturnValue((i), 2)
           Case 4
            pik(i,j) = test.ReturnValue((i), 3)
           Case 5
            pik(i,j) = test.ReturnValue((i),4)
           Case 6
              pik(i,j) = test.ReturnValue((i), 5)
           Case 7
              pik(i,j) = test.ReturnValue((i), 6)
         End Select
  next
```

```
    next

   'get niarot ereh  and value of niarot
   ' a = test.AskQuery("SELECT change ,vbuy ,value ,price ,camut ,nename
,neid , heshbonid ,update FROM NiarotEreh ORDER BY update DESC", 9)
   a = test.AskQuery("SELECT NiarotEreh.change ,NiarotEreh.vbuy
,NiarotEreh.value ,NiarotEreh.price ,NiarotEreh.camut ,NiarotEreh.nename
,NiarotEreh.neid , NiarotEreh.heshbonid ,NiarotEreh.Update FROM NiarotEreh
INNER JOIN NiarotErehMax ON NiarotEreh.Update = NiarotErehMax.Max[[[[Update ",
9)

    dim niarot(250,9)
    dim sizeNiarot
   sizeNiarot=test.GetSize
   imax=sizeNiarot-1
    for i=0 to imax
   for j=1 to 9
      Select Case (j)
         Case 1
         niarot(i,j)=test.ReturnValue((i), 0)
      Case 2
         niarot(i,j) = test.ReturnValue((i), 1)
           Case 3
         niarot(i,j) = test.ReturnValue((i), 2)
           Case 4
            niarot(i,j) = test.ReturnValue((i), 3)
           Case 5
            niarot(i,j) = test.ReturnValue((i),4)
           Case 6
              niarot(i,j) = test.ReturnValue((i), 5)
           Case 7
              niarot(i,j) = test.ReturnValue((i), 6)
     Case 8
              niarot(i,j) = test.ReturnValue((i), 7)
     Case 9
              niarot(i,j) = test.ReturnValue((i), 8)
          End Select
  next
   next
'get gemel
   a = test.AskQuery("SELECT
name,num,shovile,shoviit,hafkadot,bituah,mahut,tpa,vetek,heshbonid,update
FROM gemel", 11)
   dim gemel(250,11)
   dim sizeGemel
  sizeGemel=test.GetSize -1
       'document.write("<font>" & sizeGemel & "</font>")
       imax =sizeGemel
   for i=0 to imax
  for j=1 to 11
     Select Case (j)
        Case 1
        gemel(i,j)=test.ReturnValue((i), 0)
     Case 2
        gemel(i,j) = test.ReturnValue((i), 1)
          Case 3
        gemel(i,j) = test.ReturnValue((i), 2)
          Case 4
           gemel(i,j) = test.ReturnValue((i), 3)
          Case 5
           gemel(i,j) = test.ReturnValue((i),4)
          Case 6
             gemel(i,j) = test.ReturnValue((i), 5)
```

```
            Case 7
               gemel(i,j) = test.ReturnValue((i), 6)
         Case 8
               gemel(i,j) = test.ReturnValue((i), 7)
         Case 9
               gemel(i,j) = test.ReturnValue((i), 8)
         Case 10
               gemel(i,j) = test.ReturnValue((i), 9)
         Case 11
               gemel(i,j) = test.ReturnValue((i), 10)
           End Select
    next
     next
   'get hisahon
     a = test.AskQuery("SELECT
name,nameid,num,sdate,edate,osum,nsum,psum,Update,HeshbonID  FROM hisahon",
10)
     dim hisahon(200,10)
     sizeHisahon = test.GetSize
     imax=sizeHisahon -1
     for i=0 to imax
    for j=1 to 10
       Select Case
          Case
          hisa       )=test.ReturnValue((i), 0)
       Case 2
          hisahon(i,j) = test.ReturnValue((i), 1)
            Case 3
          hisahon(i,j) = test.ReturnValue((i), 2)
            Case 4
             hisahon(i,j) = test.ReturnValue((i), 3)
            Case 5
             hisahon(i,j) = test.ReturnValue((i),4)
            Case 6
               hisahon(i,j) = test.ReturnValue((i), 5)
            Case 7
               hisahon(i,j) = test.ReturnValue((i), 6)
       Case 8
               hisahon(i,j) = test.ReturnValue((i), 7)
       Case 9
               hisahon(i,j) = test.ReturnValue((i), 8)
       Case 10
               hisahon(i,j) = test.ReturnValue((i), 9)
           End Select
   next
    next

   test.CloseConnection
-->
</script>
</BODY>
</HTML>
```

B. Code that relates to the data.html

```
<HEAD>
<META NAME="GENERATOR" Content="Microsoft FrontPage 4.0">
<META HTTP-EQUIV="Content-type" CONTENT="text/html; charset=windows-1255">
<TITLE>Bank Account</TITLE>
<link REL="stylesheet" HREF="styles.css">
<script language="javascript">
var titleID = 'מודול חשבון בנק - תנועות בחשבון עו"ש';
```

```
</script>
</HEAD>

<body marginleft=0 margintop=0>
<script language="javascript">
parent.title.location.href='bank_title.htm'
var helpID
helpID=20;
top.hID = helpID;
parent.help.location.href='bank_tip.htm'
</script>

<div align=right>
<table border=0 cellpadding=0 cellspacing=0 width=450>
   <tr>
      <td></td>
   </tr>
   <tr align=right  valign=top>
      <td width=450>
      <script language="javascript">
      //writing tnuot table
      var i,j,k;
      document.write("<table border=1 cellpadding=0 cellspacing=0
width='450'>");
      document.write("<td align='center' bgcolor='669966'><font face='times'
size=2 color='white'><b>יתרה</td>");
      document.write("<td align='center' bgcolor='669966'><font face='times'
size=2 color='white'><b>תנועה</td>");
      document.write("<td align='center' bgcolor='669966'><font face='times'
size=2 color='white'><b>נושא</td>");
      document.write("<td align='center' bgcolor='669966'><font face='times'
size=2 color='white'><b>אסמכתא</td>");
      document.write("<td align='center' bgcolor='669966'><font face='times'
size=2 color='white'><b>תאריך</td></face>");
        for (i = 0; i <parent.data.sizeTnua; i++)
        {
        document.write("<tr>");
        for (j = 1; j < 6; j++)
           {
           document.write("<td align='right'>");
           var num=parent.data.tnua(i,j);
           if (num < 0)
              document.write("<font face='times' size=2 color=Red>" + num);
           else
              document.write("<font face='times' size=2
color=DarkSlateBlue>" + num);
           document.write(" </td>");
           }
        document.write("</tr>");
        }

     document.write("</table>");
     </script>
     </td>
  </tr>
</table>
</div>
</body>
</HTML>
```

Module 3 [Phase 1 in Fig 3a]

I. Saves data sent to the client from service provider server in HTML format (Transaction 307 in fig 3a)

```
Public Sub URLsaveAs2(strURL As String, strLocal As String)
  Dim b() As Byte
  Dim intSaveAs As Integer
  Dim strHtmlLine As String

  intSaveAs = FreeFile()
  Open strLocal For Output As #intSaveAs
  On Error GoTo ErrorHandler

   frmConnected.Inet1.RequestTimeout = 180
  b() = frmInetCtl.Inet1.OpenURL(strURL, 1)
  For t = 0 To UBound(b) - 1
    If b(t) = 10 Then
      Print #intSaveAs, strHtmlLine
      strHtmlLine = ""
    Else
      strHtmlLine = strHtmlLine & Chr(b(t))
    End If
  Next
  Print #intSaveAs, strHtmlLine
  Close #intSaveAs
  Exit Sub

ErrorHandler:
  OnError "urlsaveas2", Err, Error$, Now, "we don't have line input here"
  Exit Sub
End Sub
```

B. Example of saving data that arrived with URLSaveas2 in Database using ADO (Transaction 330 in fig 3a)

```
'*************************************************************
'DESCRIPTION : import info from heshbonall.html              *
'*and update table TnuotBank                                 *
'*************************************************************
Sub ImporttnuotHTML()
  Dim rsTnuot As ADODB.Recordset 'holds info from tnuotbank table
  Dim strNewLine As String       'input line from html file
  Dim strYearTemp   As String    'year of update
  Dim strHeshbon As String       'number of account
  Dim a As Integer               'checks if new account (a=1)
  Dim dtUpdated As Date          'updating date
  Dim fileNumber As Integer      'number of input file
  Dim dtUpdateTizmun As Date     'gets date of importing for tizmun table

  On Error GoTo error_1:
    Set rsTnuot = New ADODB.Recordset
    rsTnuot.ActiveConnection = conDB_Connection
    rsTnuot.LockType = adLockOptimistic
    rsTnuot.Open "SELECT * FROM TnuotBank", , adCmdText
    fileNumber = FreeFile()
    Open GetPath & "heshbonall.htm" For Input As #fileNumber
    Line Input #fileNumber, strNewLine
    strNewLine = TernUpperToLower(strNewLine)
    Do While Not (EOF(fileNumber))
      Do Until Mid(strNewLine, 1, 26) = "<font color=white/>|<font>" Or
EOF(fileNumber)
        Line Input #fileNumber, strNewLine
        strNewLine = TernUpperToLower(strNewLine)
```

```
        Loop
        If EOF(fileNumber) Then
          Exit Do
        End If

    'Do Until EOF(fileNumber)
      'MsgBox ("strNewLine=" & Mid(strNewLine, 98, 8))
      'Line Input #fileNumber, strNewLine
    'Loop
      dtUpdateTizmun = Mid(strNewLine, 98, 8)
      Line Input #fileNumber, strNewLine
      dtUpdated = Mid(strNewLine, 77, 8)
      strYearTemp = Mid(strNewLine, 83, 2)
      Line Input #fileNumber, strNewLine
      strHeshbon = Mid(strNewLine, 71, 9)
      Line Input #fileNumber, strNewLine
      Line Input #fileNumber, strNewLine
      Line Input #fileNumber, strNewLine
      strNewLine = TernUpperToLower(strNewLine)
      Do While Mid(strNewLine, 1, 26) = "<font color=white/>|<font>"
        If (Mid(strNewLine, 80, 1) = "/") And (Mid(strNewLine, 68, 8<> (
("בתוקף עד"Then
          rsTnuot.AddNew
          If Mid(strNewLine, 27, 13) = "             " Then
          Else
            rsTnuot("itra") = Mid(strNewLine, 29, 13)
          End If
          If Mid(strNewLine, 41, 5) = "     " Then
          Else
            rsTnuot("datev") = Mid(strNewLine, 41, 5)
          End If
          rsTnuot("value") = IIf(Mid(strNewLine, 54, 4) <> "    ",
Mid(strNewLine, 50, 16), Mid(strNewLine, 60, 16))
          If Mid(strNewLine, 54, 4) = "    " Then rsTnuot("value") = -1 *
rsTnuot("value")
          rsTnuot("date") = Mid(strNewLine, 78, 2) & "/" & Mid(strNewLine,
81, 2) & "/" & strYearTemp
          rsTnuot("asmachta") = Mid(strNewLine, 84, 8)
          rsTnuot("makor") = Trim(Mid(strNewLine, 93, 14))
          rsTnuot("heshbonid") = strHeshbon
          rsTnuot("update") = dtUpdated
          rsTnuot.Update
        End If
        Line Input #fileNumber, strNewLine
        strNewLine = TernUpperToLower(strNewLine)
      Loop
    Loop
    Close #fileNumber
    rsTnuot.Close
    UpdateTizmun "tnuotbank", dtUpdateTizmun
  Exit Sub
error_1:
  OnError "ImporttnuotHTML", Err, Error$, dtUpdated, strNewLine
   Resume Next
End Sub
```

C.
Updatedata - a module that uses the above 2 functions (A & B) in order to login into and get personal data from service provider server and save it in the local database.

After updating the data the procedure calls procedure CheckRedFlags (282 in fig 2b) which accesses the DMS in order to analyze the data in database and if relevant, shows enriched data.

```
Public Sub UpdateData()
URLsaveAs2
"http://hb.bankleumi.co.il:8000/homebank/FirstLoginMiddleFrame.asp?/L=H/TBL/U
=I127011/A=2", AppPath & "hb.html"
URLsaveAs2
"http://hb.bankleumi.co.il:8000/Homebank/HBIsapi.dll?MfcISAPICommand=GetForm&
query=/L=H/S=01/T=01/Q=01/AP1=*/WCQUERY/IE4/STD/U=I127011/A=2", AppPath &
"hb.htm"
URLsaveAs2
"http://hb.bankleumi.co.il:8000/Homebank/HBIsapi.dll?MfcISAPICommand=GetForm&
query=/L=H/S=01/T=01/Q=01/AP1=678330002567212/WCQUERY/IE4/STD/U=I127011/A=2",
AppPath & "heshbon.htm"
ImporttnuotHTML
CheckRedFlags
End sub
```

D. The following function periodically updates the client's database with data from the service provider server, according to the relevance of changes in the personal data.

```
  Dim rsTizmun As ADODB.Recordset
'  open table tizmun
  Set rsTizmun = New ADODB.Recordset
    rsTizmun.ActiveConnection = conDB_Connection
    rsTizmun.LockType = adLockOptimistic
    rsTizmun.Open "SELECT tablename,update FROM Tizmun", , adCmdText
'  find the requiered record  and update the updating date
  Do Until (rsTizmun.EOF)
    If (rsTizmun("tablename") = tableName) Then
      rsTizmun("update") = newDate
      rsTizmun.Update
      Exit Do
    Else
      rsTizmun.MoveNext
    End If
  Loop
  rsTizmun.Close
End Sub



'*************************************************************
'*DESCRIPTION : this function gets a table ID number    *
'*             and activates the function that          *
'*             updates this table                       *
'*INPUT :      : integer - table id number              *
'*************************************************************
Sub ActivateFunction(ByVal tableName As String)
  Select Case (tableName)
    Case "pikdonot"
      ImportPikadonAll

    Case "gemel"
      ImportGemel

    Case "hisahon"
      ImportHisahon
```

```
      Case "niarotereh"
        UpdateNiarotEreh

      Case "tnuotbank"
        blnNeedImportOld = True
        ImporttnuotHTML
        LocateTnua

      Case "visaold"
        Importtnuotv

      Case "visanew"
        ImporttnuotvNew

      Case "tnuotold"
        ImportTnuotOld

    End Select
End Sub
*************************************************************
*'DESCRIPTION: this function checks if the tables that  *
        *'need to be updated each month were       *
        *'updated in the current month, if they    *
        *'weren't ,the function updates them here  *
*************************************************************
Sub UpdateAllTables()
  Dim rsTizmun As ADODB.Recordset 'get info from tizmun table
  Dim strUpdated As String        'get date of updating
  Dim intHefresh As Integer       'hefresh between the date and updating date
  Dim strPath As String           'path of application
  blnImportFailed = False        'assume import succedes
  strPath = GetPath
  URLsaveAs1 "http://hb.bankleumi.co.il:8000/homebank/default.asp", strPath &
"file1.html"
  URLsaveAs1
"http://hb.bankleumi.co.il:8000/homebank/top.asp?/L=H/TBL/U=I127011/A=2",
strPath & "file2.html"
  URLsaveAs1
"http://hb.bankleumi.co.il:8000/homebank/FirstLoginMiddleFrame.asp?/L=H/TBL/U
=I127011/A=2", strPath & "file3.html"
  URLsaveAs1
"http://hb.bankleumi.co.il:8000/homebank/HBIsapi.dll?MfcISAPICommand=GetForm&
query=/L=H/TBL/U=I127011/A=2", strPath & "file4.html"
  URLsaveAs1
"http://hb.bankleumi.co.il:8000/homebank/Alert.asp?/L=H/TBL/U=I127011/A=2",
strPath & "file5.html"
  URLsaveAs1
"http://hb.bankleumi.co.il:8000/homebank/ToolBoxAfterNew.asp?/L=H/TBL/U=I1270
11/A=2", strPath & "file6.html"
  URLsaveAs1
"http://hb.bankleumi.co.il:8000/homebank/HBIsapi.dll?MfcISAPICommand=ShowButt
ons&command=/L=H/TBL/U=I127011/A=2", strPath & "file7.html"
  URLsaveAs1
"http://hb.bankleumi.co.il:8000/homebank/Adv.asp?/L=H/TBL/U=I127011/A=2",
strPath & "file8.html"
  URLsaveAs1 "http://hb.bankleumi.co.il:8000/homebank/default.asp", strPath &
"file9.html"


  UpdateTnua                      'update tnuotbank and check for other
                    updatings
```

```
    If (blnImportFailed = True) Then
      MsgBox ("ImportFailed , try later")
      Exit Sub
    End If

  '  get info from tizmun table
    Set rsTizmun = New ADODB.Recordset
      rsTizmun.ActiveConnection = conDB_Connection
  '    rsTizmun.LockType = adLockOptimistic
      rsTizmun.Open "SELECT tablename,filename,update,Tkufa,UrlPath FROM
 Tizmun", , adCmdText
      rsTizmun.MoveFirst

  '  if table is updated each month (tkufa=1):
  -    '  check if table needs to be updated this month
  '  if table is updated each week   (tkufa=2):
  -    '  check if table needs to be updated this week
  '  if table is updated each day    (tkufa=3):
  -    '  update table
  '  URLsaveAs1
 "http://hb.bankleumi.co.il:8000/homebank/FirstLoginMiddleFrame.asp?/L=H/TBL/U
 =I127011/A=2", AppPath & "hb.html"
    Do Until (rsTizmun.EOF)
      strUpdated = rsTizmun("update")
      Select Case (rsTizmun("Tkufa"))
        Case 1       'update each month
          If (Month(strUpdated) <> Month(date)) Then
            ImportAndUpdate rsTizmun("UrlPath"), rsTizmun("FileName"),
 rsTizmun("TableName")
          End If

        Case 2       'update each week
          hefresh = DateDiff("d", strUpdated, date)
          If hefresh >= 7 Then
            ImportAndUpdate rsTizmun("UrlPath"), rsTizmun("FileName"),
 rsTizmun("TableName")
          Else
            If Weekday(strUpdated) > Weekday(date) Then
              ImportAndUpdate rsTizmun("UrlPath"), rsTizmun("FileName"),
 rsTizmun("TableName")
            End If
          End If

        Case 3      'update each day
          If (rsTizmun("TableName") = "tnuotold") Then
            If blnNeedImportOld = True Then 'update tnuotold only if nessesary
              ImportAndUpdate rsTizmun("UrlPath"), rsTizmun("FileName"),
 rsTizmun("TableName")
            End If
          Else
            If (rsTizmun("TableName") <> "tnuotbank") Then
              ImportAndUpdate rsTizmun("UrlPath"), rsTizmun("FileName"),
 rsTizmun("TableName")
            End If
          End If
      End Select
      rsTizmun.MoveNext
    Loop
    rsTizmun.Close
    MsgBox ("finished updating all")
End Sub
```

```
Sub ImportAndUpdate(ByVal urlPath As String, ByVal fileName As String, ByVal
tableName As String)
  Dim strFullName As String           'name of file +path
  strFullName = GetPath() & fileName
  If (ImportFile(urlPath, strFullName) = True) Then
    ActivateFunction (tableName)
  Else
    blnImportFailed = True
  End If
End Sub
Sub UpdateTnua()
  Dim rsTizmun As ADODB.Recordset 'holds records of tizmun table
  Dim strUrlPath As String    'url of tnuot html
  Dim strFileName As String   'name of file to save
  Dim strTableName As String  'name of record in table tizmun

  Set rsTizmun = New ADODB.Recordset
  rsTizmun.ActiveConnection = conDB_Connection
  rsTizmun.LockType = adLockOptimistic
  rsTizmun.Open "SELECT tablename,update,filename,Tkufa,UrlPath FROM Tizmun",
, adCmdText

  Do Until (rsTizmun("tablename") = "tnuotbank")
    rsTizmun.MoveNext
  Loop
  strUrlPath = rsTizmun("UrlPath")
  strFileName = rsTizmun("FileName")
  strTableName = rsTizmun("TableName")
  rsTizmun.Close

  ImportAndUpdate strUrlPath, strFileName, strTableName

End Sub
```

Module 3 - [phase 2 of fig 3a]
UpdateApp - Gets general data and interface objects (HTML) from Server (306), and saves it in the client's machine on a database (320). The function is an example of transaction 312, 332, 322 in fig 3a.

```
Public Sub UpdateApp()
  Dim oldDate, newDate As Variant
  Dim strNewFileName, strOldFileName As String
  Dim lngNewFileVersion, lngOldFileVersion  As Long

  If blnCharLoaded = True Then
    Character.MoveTo 366, 97
    Character.Speak "I am now searching for newer versions of B-Midas
pages..."

    Character.Play "search"
  End If

  frmUpdateApp.Show
  UrlSaveAs Homepage + "NewVersionInfo.txt", AppPath & "NewVersionInfo.txt"
  intOld = FreeFile()
  Open AppPath + "VersionInfo.txt" For Input As #intOld
  intNew = FreeFile()
  Open AppPath + "NewVersionInfo.txt" For Input As #intNew

  Do While (Not EOF(intNew)) Or Not EOF(intOld) ' Loop until end of a file
    Input #intOld, strOldFileName, oldDate
```

```
      Input #intNew, strNewFileName, newDate

'      Add file
      Do While (strNewFileName < strOldFileName) And (Not EOF(intNew))
        frmUpdateApp.List1.AddItem "Adding " + Replace(strNewFileName, "/",
"\")
        URLsaveAs2 Homepage + strNewFileName, AppPath + Replace(strNewFileName,
"/", "\")
        Input #intNew, strNewFileName, newDate
      Loop

'      Delete file
      Do While (strOldFileName < strNewFileName) And EOF(intOld)
        frmUpdateApp.List1.AddItem "Deleting " + Replace(strOldFileName, "/",
"\")
        Kill AppPath + strOldFileName
        Input #intOld, strOldFileName, oldDate
      Loop

'      Update file
      If strOldFileName = strNewFileName Then
        If oldDate < newDate Then
          frmUpdateApp.List1.AddItem "Updating " + Replace(strNewFileName, "/",
"\")
          Beep
          URLsaveAs2 Homepage + strNewFileName, AppPath +
Replace(strNewFileName, "/", "\")
        End If
      End If
    Loop
    Close #intOld
    Close #intNew
    FileCopy AppPath + "NewVersionInfo.txt", AppPath & "VersionInfo.txt"
    Kill AppPath + "newversioninfo.txt"
End Sub
```

Module 4 - [Part of Personal Agent] Example of functions used to access database through the DMS (332 in fig 3a) in order to analyze the personal data and tag it relevant or irrelevant and show it enriched.

```
Dim db_file As String
Public conn As ADODB.Connection
Dim rs As ADODB.Recordset
Dim rs1 As ADODB.Recordset
Dim rs2 As ADODB.Recordset
Public rs3 As ADODB.Recordset
Dim rs4 As ADODB.Recordset
Dim txt As String
Dim fld As Field
Dim fld1 As Field
Dim fld2 As Field
Dim arr() As tnua
Dim arr1() As visa
Dim arr2() As pik
Dim arr3() As itra

Private Type tnua
  itra As String
  value As String
  makor As String
  asmachta As String
  date As String
```

```
End Type
Private Type visa
  bdate As String
  esek As String
  bsum As String
  asmachta As String
  hsum As String
End Type
Private Type pik
  osum  As String
  nsum  As String
  heshbonid As String
  Update  As String
  name   As String
  sdate  As String
  ribit As String
End Type
Private Type itra
  heshbonname As String
  heshbonid As String
  firstofitra As String
  maxofdate As String
End Type
Private Type rasconn
  dwSize As Long
  hrasconn As Long
  szEntryName As String * 257
  szDeviceType As String * 17
  szDeviceName As String * 130
End Type

Public Sub init()
'  Get the data.
  db_file = App.Path ' & "\bank\data\"

  If Right$(db_file, 1) <> "\" Then db_file = db_file & "\"
  db_file = db_file & "DB.mdb"
'MsgBox ("path= " & db_file)
'  Open a connection.
  Set conn = New ADODB.Connection
  conn.ConnectionString = _
"    Provider=Microsoft.Jet.OLEDB.3.51;" & _
"    Data Source=" & db_file & ";" & _
"    Persist Security Info=False"
  conn.Open
End Sub

Public Function GetTnua(X As String) As String
    Dim d As tnua
    ReDim arr(0)
    If (X = "") Then
      Set rs = conn.Execute("SELECT Itra , Value , Makor , Asmachta , Date
FROM TnuotBank ORDER BY Date DESC", , adCmdText)
    Else
      Set rs = conn.Execute("SELECT Itra , Value , Makor , Asmachta , Date
FROM TnuotBank WHERE HeshbonID='" & X & "' ORDER BY Date DESC  ", ,
adCmdText)
    End If
    If rs.EOF Then
      GetTnua = Null
    Else
      txt = ""
```

```
        For I = 0 To 9
'          txt = txt & Trim$(fld.value)
          d.itra = rs.Fields(0)
          d.value = rs.Fields(1)
          d.makor = rs.Fields(2)
          d.asmachta = rs.Fields(3)
          d.date = rs.Fields(4)
          arr(I) = d
          ReDim Preserve arr(UBound(arr) + 1)
          rs.MoveNext
        Next I
'       If Len(txt) > 0 Then txt = Left$(txt, Len(txt) - 1)
        GetTnua = " "
      End If
End Function
Public Function GItra() As String
   Set rs3 = conn.Execute(" SELECT Max(TnuotBank.Date) AS MaxOfDate,
First(TnuotBank.Itra) AS FirstOfItra, TnuotBank.HeshbonID,
HeshBank.HeshbonName FROM TnuotBank INNER JOIN HeshBank ON
TnuotBank.HeshbonID = HeshBank.HeshbonID GROUP BY TnuotBank.HeshbonID,
HeshBank.HeshbonName Having (First(TnuotBank.itra) <> 0) ORDER BY
Max(TnuotBank.Date) DESC ", , adCmdText)
   Dim e As itra
   Dim p As Integer
   ReDim arr3(0)
   p = 0
   If rs3.EOF Then
      GItra = ""
   Else
     Do Until (rs3.EOF)
       e.maxofdate = rs3.Fields(0)
       e.firstofitra = rs3.Fields(1)
       e.heshbonid = rs3.Fields(2)
       e.heshbonname = rs3.Fields(3)
       arr3(p) = e
       p = p + 1
       ReDim Preserve arr3(UBound(arr3) + 1)
'         txt = txt & Trim$(fld1.value) & vbTab
       rs3.MoveNext
     Loop
'        If Len(txt) > 0 Then txt = Left$(txt, Len(txt) - 1)
'        rs1.MoveNext
        GItra = ""
  End If
End Function
Public Function Gvaltnua(X As Integer, Y As String) As String
  Select Case (Y)
    Case "itra"
      Gvaltnua = arr(X).itra
    Case "value"
      Gvaltnua = arr(X).value
    Case "makor"
      Gvaltnua = arr(X).makor
    Case "asmachta"
      Gvaltnua = arr(X).asmachta
    Case "date"
      Gvaltnua = arr(X).date
  End Select
End Function
Public Function Grecsnum(X As String) As String
  Select Case (X)
  Case "tnua"
```

```
      Grecsnum = UBound(arr())
    Case "visa"
      Grecsnum = UBound(arr1())
    Case "pik"
      Grecsnum = UBound(arr2())
    Case "itra"
      Grecsnum = UBound(arr3())
    End Select
End Function
Public Function Gvalitra(X As Integer, Y As String) As String
  Select Case (Y)
    Case "heshbonname"
      Gvalitra = arr3(X).heshbonname
    Case "heshbonid"
      Gvalitra = arr3(X).heshbonid
    Case "firstofitra"
      Gvalitra = arr3(X).firstofitra
    Case "maxofdate"
      Gvalitra = arr3(X).maxofdate
  End Select
End Function

Public Sub destructor()
' rs.Close
' rs1.Close
' rs2.Close
' rs3.Close
  rs4.Close
  conn.Close
End Sub
```

Module 5 - [Phase3, 311 in fig 3a]
a. An example of a way of sending enriched object to client in a form of an E-Mail message the personal agent sends to the client.

```
Private Sub SendEmailRedFlags()
  Dim intFileNum As Integer
  Dim TheOutlook, TheMapiName, TheMail

  Set TheOutlook = CreateObject("Outlook.Application")
  Set TheMapiName = TheOutlook.GetNamespace("MAPI")
    If TheOutlook = "Outlook" Then
      TheMapiName.Logon "profile", "password"
      Set TheMail = TheOutlook.CreateItem(0)
      TheMail.To =  EmailAddress
      TheMail.Subject הודעה מהסוכן החכם של" = B-Midas"
      TheMail.Body"500- הנך צפוי ליתרה שלילית של "
      TheMail.Send
    End If
End Sub
```

b. An example of a way of sending enriched object to client in a form of an Cellular SMS (Short Messaging System) message the personal agent sends to the client.

This function receives a message to be sent, identifies the preferred cellular SMS system, and activates the relevant function for that service provider. (In our example UpdateCellLog)

```
Sub AlertCell(ByVal strAlert As String)
    Dim strPath As String 'aplication path
```

```
    strMessage = strAlert
    strPath = App.Path & "\"
    strAreaCode = GetSetting("takzibit", "Init", "CellAreaCode", "")
                        'the user has cellcom
    If strAreaCode = "052" Or strAreaCode = "053" Or strAreaCode = "058" Then
        UpdateCellLog          'insert to cellLog.htm
                            'the apropriate username and password

                        'we want to give time to load the page
                        'and only after get session id
        frmCellSend.Timer1.Enabled = True
                        'opening cellcom login page
        frmCellSend.WebBrowser1.Navigate strPath & "cellLog.htm"
                'cellcom send message page is being opened by
                'cellLog.htm ,in order to read session id number
   Else         'the user has pelephone
        UpdatePele
        frmCellSend.Timer1.Enabled = False
        frmCellSend.WebBrowser1.Navigate strPath & "Peletext Sirvice.htm"
   End If
End Sub
```

This function activates two HTML pages the first log.html & message.html. This function enriches log.html with the users logon name and password and submits the html object to the service providers' server. The service providers' server returns an html object and the function the activates function: insertnumber

```
Sub UpdateCellLog()
    Dim intFileInput As Integer 'number of input file
    Dim intFileOutput As Integer 'number of temp output file
    Dim strUserName As String  'username
    Dim strPassWord As String  'password
    Dim strLine As String 'line input


    strUserName = GetSetting("takzibit", "Init", "CellUserName", "bmidas")
    strPassWord = GetSetting("takzibit", "Init", "CellPassWord", "bmidas")

   intFileInput = FreeFile()
   Open App.Path & "\cellLogTavnit.htm" For Input As #intFileInput
   intFileOutput = FreeFile()
   Open App.Path & "\cellLog.htm" For Output As #intFileOutput

  Do Until EOF(intFileInput)
    Line Input #intFileInput, strLine
    If (strLine Like "*<!--insert username here-->*") Then
        strLine = Replace(strLine, "<!--insert username here-->",
strUserName)
    End If
    If (strLine Like "*<!--insert password here-->*") Then
        strLine = Replace(strLine, "<!--insert password here-->",
strPassWord)
    End If
    Print #intFileOutput, strLine
  Loop
  Close #intFileInput
  Close #intFileOutput
End Sub
```

This function fills the html object received by the server with the message and submits it to the server. The submission of message.html sends the message to the user.

```
'*****************************************
'*DESCRIPTION: this function gets a  number *
'*             and inserts it to the right  *
'*             places in cell hodaot file   *
'*INPUT:       strNumber - string           *
'*****************************************
Sub InsertNumber(strSessionNumber As String)
    Dim intFileInput As Integer  'input file number
    Dim intFileOutput As Integer 'output file number
    Dim strLine As String        'line input

    strNumber = GetSetting("takzibit", "Init", "CellPhoneNumber", "498164")
    strAreaCode = GetSetting("takzibit", "Init", "CellAreaCode", "053")
    strCounter = 100 - Len(strMessage)
                    'openning טלקום הודעות
                    'and writing session number
    intFileInput = FreeFile()
    Open App.Path & "\cellHodTavnit.htm" For Input As #intFileInput
    intFileOutput = FreeFile()
    Open App.Path & "\cellHod.htm" For Output As #intFileOutput

    strLine = ""
    Do Until EOF(intFileInput)
           If EOF(intFileInput) Then
               Exit Do
           End If
           Line Input #intFileInput, strLine
           If (strLine Like "*<!--write your session number here-->*") Then
              strLine = Replace(strLine, "<!--write your session number
here-->", strSessionNumber)
           End If
           If (strLine Like "*<!--insert area code here-->*") Then
                strLine = Replace(strLine, "<!--insert area code here-->",
strAreaCode)
           End If
           If (strLine Like "*<!--insert number here-->*") Then
                strLine = Replace(strLine, "<!--insert number here-->",
strNumber)
           End If
           If (strLine Like "*<!--insert counter here-->*") Then
                strLine = Replace(strLine, "<!--insert counter here-->",
strCounter)
           End If
           If (strLine Like "*<!--insert msg here-->*") Then
                strLine = Replace(strLine, "<!--insert msg here-->",
strMessage)
           End If
        Print #intFileOutput, strLine
    Loop
    Close #intFileInput
    Close #intFileOutput
    strPath = App.Path & "\"
    frmCellSend.WebBrowser1.Navigate strPath & "cellHod.htm"
End Sub

Log.html
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<!-- saved from
url=(0054)http://text.cellcom.co.il/webp/cgi/pu/pu_login.exe?x=x -->
<HTML><HEAD><TITLE>טלקום טקסט</TITLE>
```

```
<META content="text/html; charset=iso-8859-8"
http-equiv=Content-Type><!--<META HTTP-EQUIV="Content-Type"
content="text/html; charset=iso-8859-8">-->
<script language=javascript>
function fillSubmit()
{

document.LoginForm.ShownUserName.value =  "דורון"
document.LoginForm.ShownPassword.value = "גיאת"
CheckLoginSubmit()
document.LoginForm.submit();
}
</script>


<SCRIPT src="סלקום טקסט_files/Resize.js"></SCRIPT>

<SCRIPT src="סלקום טקסט_files/GeneralFunctions.js"></SCRIPT>

<SCRIPT src="סלקום טקסט_files/PU_Login.js"></SCRIPT>

<META content="MSHTML 5.00.2314.1000" name=GENERATOR></HEAD>
<BODY onload ="fillSubmit()" bgColor=#f4efdd link=black vLink=black>
<DIV align=center>
<SCRIPT>
      sMissingUserName = "הקש את שם המשתמש שלך בבקשה";
      sMissingPassword = "הקש סיסמה בבקשה";
      bAlreadySubmitted = false;
</SCRIPT>

<TABLE align=center border=0 cellPadding=0 cellSpacing=0><!-- Advertisment
---------------------------------------------------------------------->
  <TBODY>
  <TR>
    <TD align=right colSpan=5 vAlign=top><A
      href="http://www.cellcom.co.il/framemain8.html"><IMG border=0 height=55
      src="סלקום טקסט_files/Banner.gif" width=400></A></TD>
    <TD align=left vAlign=top><A href="http://www.cellcom.co.il/"><IMG
      border=0 height=66 src="סלקום טקסט_files/CellcomLogo.gif"
  width=141></A></TD></TR><!-- Navigation Bar
----------------------------------------------------------------------->
  <TR>
    <TD align=right colSpan=6 vAlign=bottom><A
      href="http://www.cellcom.co.il/framemain4.html"><IMG border=0 height=46
      src="סלקום טקסט_files/CellcomTextLogo2.gif" width=141></A></TD></TR>
  <TR>
    <TD align=right colSpan=5 vAlign=bottom><IMG height=18
      src="סלקום טקסט_files/tit_Login.gif" width=471></TD>
    <TD align=middle bgColor=#ad0000><A
      href="http://192.115.11.18/webp/Cgi/PU/PU_Login.exe?Lang=Eng"><IMG
      border=0 height=21 src="סלקום טקסט_files/btn_English.gif"
  width=86></A></TD></TR>
  <TR>
    <TD align=right colSpan=5 vAlign=top>
      <TABLE bgColor=#fbae07 border=0 cellPadding=0 cellSpacing=0 height=350
      width=471>
        <TBODY>
        <TR>
          <TD align=middle vAlign=top>
            <TABLE bgColor=white border=0 cellPadding=3 cellSpacing=0
height=345
            width=467>
```

```
                <TBODY>
                <TR>
                  <TD align=right colSpan=3><!--FONT FACE="Courier New
(Hebrew)" SIZE="2" COLOR="#AD0000"--><FONT
                    color=#ad0000 size=2><B>  </B></FONT></TD></TR>
                <TR>
                  <TD colSpan=3>
                    <HR>
                  </TD></TR>
                <TR>
                  <TD align=left vAlign=bottom><A

href="http://192.115.11.18/webp/Cgi/PU/PU_RegForm.exe?Lang=Heb"><IMG
                    border=0 height=34 src="סלקום
טקסט_files/btn_Registration.gif"
                    width=69></A></TD>
                  <TD align=right colSpan=2 vAlign=top><FONT size=2><B>שדה
                    שמתשמ</B></FONT><BR><FONT color=#ad0000 size=2>תוריש יינמל
                    .סוקלס תשרב 'טסקט סוקלס <BR>תובותכ תועדוה חולשמ רשפאמ הז תורש
                    .ינורטקלא ראודב המסיסה רילא חלשית המשרהה רחאל<BR>
</FONT></TD></TR>
                <FORM action=http://192.115.11.18/webp/Cgi/PU/PU_CheckLogin.exe
                method=post name=LoginForm
                onsubmit="return CheckLoginSubmit();"><INPUT name=Lang
type=hidden
                value=Heb>
                <TR>
                  <TD colSpan=3>
                    <HR>
                  </TD></TR>
                <TR>
                  <TD align=right colSpan=3><FONT size=2><B>סושר
                    שמתשמ</B></FONT><BR><FONT color=#ad0000 size=2>:תורישל
הסינכל
                    סיאבה סיטרפה תא אלמל אנ </FONT></TD></TR>
                <TR>
                  <TD align=left><A href="javascript:ResetLoginForm();"><IMG
                    border=0 height=32 src="סלקום טקסט_files/btn_Reset.gif"
                    width=51></A></TD>
                  <TD align=right><FONT face="Courier New (Hebrew)"
                    size=3><INPUT maxLength=50 name=ShownUserName size=15>
<INPUT
                    name=UserName type=hidden> </FONT></TD>
                  <TD align=right><!--FONT FACE="Courier New (Hebrew)"
SIZE="2"--><FONT
                    size=2><B>:שם שמתשמ</B> </FONT></TD></TR>
                <TR>
                  <TD align=left><INPUT border=0 height=32
                    src="סלקום טקסט_files/btn_Login.gif" type=image
width=51></TD>
                  <TD align=right><FONT face="Courier New (Hebrew)"
                    size=3><INPUT maxLength=50 name=ShownPassword size=15
                    type=password> <INPUT name=PASSWORD type=hidden>
</FONT></TD>
                  <TD align=right><!--FONT FACE="Courier New (Hebrew)"
SIZE="2"--><FONT
                    size=2><B>:המסים</B> </FONT></TD></TR>
                <TR>
                  <TD align=right colSpan=3>
                    <HR>
                    <FONT color=#ad0000 size=2>,הסינכה תמסים תא רכוז דניאו רבעב
```

```
                    תמשרנ סא <BR><FONT color=#ad0000 size=2>.ינורטקלא ראוד
תועצמאב
                    בוש חלשת המסיסו <FONT color=black size=2><A

href="http://192.115.11.18/webp/Cgi/PU/PU_SendNewPwdForm.exe?Lang=Heb">אכ]
                    קחל</A>

</FONT></FONT></FONT></TD></TR></TBODY></TABLE></TD></TR></TBODY></TABLE></TD
>
    <TD bgColor=#ad0000> </TD></TR>
  <TR>
    <TD align=right colSpan=6 vAlign=top><IMG height=58
      src="סלקום טקסט_files/sbr_Bottom.gif" width=141></TD></TR>
  <SCRIPT>
var sCookieValue = "";
var sTempCookieValue = GetCookie("PU_CellcomTextLogin");
sCookieValue = sCookieValue+sTempCookieValue;
if (sCookieValue != "null")
{
      asDetails = sCookieValue.split(";");
      document.LoginForm.ShownUserName.value = "";
      document.LoginForm.ShownPassword.value = "";
}
</SCRIPT>
  </FORM></TBODY></TABLE><!--</DIV>
</BODY>
</HTML>--><!--<A HREF="aaa">
<IMG  SRC="/webp/Images/Pu/Heb/General/Footer.gif"
            BORDER="0"
            ISMAP>
</A>--><IMG
border=0 height=47 src="סלקום טקסט_files/Footer.gif" useMap=#FooterMap
width=619> <MAP name=FooterMap><AREA coords=320,0,375,24
  href="http://www.cellcom.co.il/" shape=RECT target=Cellcom
  title="Cellcom home page"><AREA coords=64,27,108,47
  href="http://www.box.co.il/" shape=RECT target=Box title="Box home
page"><AREA
  coords=552,27,619,47 href="http://www.netology-sms.com/" shape=RECT
  target=Netology title="Netology home page"></MAP></DIV></BODY></HTML>

Message.html
<HTML>
<HEAD>
<!--<META HTTP-EQUIV="Content-Type" content="text/html;
charset=iso-8859-8">-->
<META HTTP-EQUIV="Pragma" CONTENT="no-cache">
<TITLE>סלקום טקסט - משלוח הודעות</TITLE>
<SCRIPT SRC="סלקום טקסט - משלוח הודעות_files/Resize.js"></SCRIPT>
<SCRIPT SRC="סלקום טקסט - משלוח הודעות_files/GeneralFunctions.js"></SCRIPT>
<SCRIPT SRC="סלקום טקסט - משלוח הודעות_files/PU_RegFunctions.js"></SCRIPT>
<SCRIPT SRC="סלקום טקסט - משלוח הודעות_files/SetDbInfo.js"></SCRIPT>
<SCRIPT SRC="סלקום טקסט - משלוח הודעות_files/MessageCounter.js"></SCRIPT>
<SCRIPT SRC="סלקום טקסט - משלוח הודעות_files/SubmitMessage.js"></SCRIPT>
<SCRIPT>
nPhoneLen               = 6;
nMaxMsgLen              = 100;
sAreaCodeAlert                  = ".נא לבחור קידומת";
sPhoneNumberAlert1              = "עליך להקיש מספר טלפון תקני בן";
sPhoneNumberAlert2              = "ספרות";
```

```
sAllowed                =
"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890`~!@^*()-_:;.7
? ";
sHebrew                         = "אבגדהוזחטיכלמנסעפצקרשתךםןץף";
sAreaCodes              = "052;053;058";
sUrgencies              = "רגיל;דחוף;חרום";
nDefaultUrgency                 = "0";
sMissingSenderAlert             = "הקש את שמך בבקשה";
sIllegalSenderAlert             = "הוא תו לא חוקי בשם השולח";
sMissingMsgAlert        = "הקש הודעה בבקשה";
sIllegalMsgAlert        = "הוא תו לא חוקי בהודעה";

// General Alerts
sMissingAlert                   = "הקש בבקשה";
sIllegalCharAlert       = "הוא תו לא חוקי ב";
sLeadingBlankAlert              = "תו רווח אינו חוקי בתחילת";

// Fields types
sSenderFieldType        = "שם שולח";
sRcptFieldType                  = "";
sGrpFieldType                   = "";
sMsgFieldType                   = "הודעה";
sNickNameFieldType              = "שם משתמש";
sFullNameFieldType              = "";
sPwdFieldType                   = "סיסמת משתמש";
sOrgFieldType                   = "";
sDefaultFieldType       = "שדה הנתונים המסומן";
</SCRIPT>
<script language=javascript>
function fillForm()
      {
      document.MinSendForm.SenderName.value = "bmidas"
      document.MinSendForm.RcptAreaCode.value ="<!--insert area code here-->"
      document.MinSendForm.RcptNumber.value="<!--insert number here-->"
      document.MinSendForm.CurrMsg.value ="<!--insert msg here-->"
      document.MinSendForm.Counter.value="<!--insert counter here-->"
      CheckPuSendSubmit('Heb');
      document.MinSendForm.submit();
      }
</script>

</HEAD>

<BODY BGCOLOR="#F4EFDD"
            onLoad='SetDocumentCapture(   document.MinSendForm.CurrMsg,

      document.MinSendForm.Counter);
                        UpdateCounter(    document.MinSendForm.CurrMsg,

      document.MinSendForm.Counter);
                                                fillForm();'
            onUnload=' SetDocumentRelease(); ' >
<!--<BODY   onLoad = 'fillForm()' BGCOLOR="#F4EFDD">-->
<DIV ALIGN="Center">
<TABLE BORDER="0" CELLPADDING="0" CELLSPACING="0" ALIGN="Center">

<!-- Advertisment
------------------------------------------------------------------------>
<TR>
```

```
      <TD ALIGN="Right" VALIGN="Top" COLSPAN="5"><A
HREF="http://www.cellcom.co.il:80/framemain3.html"><IMG SRC="סלקום טקסט - משלוח
הודעות_files\Banner.gif" BORDER="0" WIDTH="400" HEIGHT="55"></A></TD>

      <TD ALIGN="Left" VALIGN="Top"><A HREF="http://www.cellcom.co.il"><IMG
SRC="סלקום טקסט - משלוח הודעות_files\CellcomLogo.gif" BORDER="0" WIDTH="141"
HEIGHT="66"></A></TD>
</TR>

<!-- Navigation Bar
------------------------------------------------------------------------>
<TR>
      <TD ALIGN="Right" VALIGN="Bottom"><A
HREF="http://192.115.11.18/webp/Cgi/PU/PU_Logout.exe?UserName=bmidas&Lang=Heb
"><IMG SRC="סלקום טקסט - משלוח הודעות_files.gif" BORDER="0" WIDTH="84"
HEIGHT="59"></A></TD>

      <!--<TD ALIGN="Right" VALIGN="Bottom"><A
HREF="http://192.115.11.18/webp/Cgi/PU/PU_Help.exe?UserName=bmidas&SessionID=
<!--write your session number here-->

&Lang=Heb"><IMG SRC="סלקום טקסט - משלוח הודעות_files\ico_Help-Y.gif" BORDER="0"
WIDTH="94" HEIGHT="59"></A></TD>-->
      <TD ALIGN="Right" VALIGN="Bottom"><IMG SRC="סלקום טקסט - משלוח
הודעות_files\ico_Help-Y.gif" BORDER="0" WIDTH="94" HEIGHT="59"></TD>

      <TD ALIGN="Right" VALIGN="Bottom"><IMG SRC="סלקום טקסט - משלוח
הודעות_files\ico_EmptyBar.gif" BORDER="0" WIDTH="94" HEIGHT="59"><IMG
SRC="סלקום טקסט - משלוח הודעות_files\ico_Filler.gif" BORDER="0" WIDTH="3"
HEIGHT="18"><IMG SRC="סלקום טקסט - משלוח הודעות_files\ico_Filler.gif"
BORDER="0" WIDTH="3" HEIGHT="18"><IMG SRC="סלקום טקסט - משלוח
הודעות_files\ico_Filler.gif" BORDER="0" WIDTH="3" HEIGHT="18"><IMG SRC="סלקום
טקסט - משלוח הודעות_files\ico_Filler.gif" BORDER="0" WIDTH="3" HEIGHT="18"><IMG
SRC="סלקום טקסט - משלוח הודעות_files\ico_Filler.gif" BORDER="0" WIDTH="3"
HEIGHT="18"><IMG SRC="סלקום טקסט - משלוח הודעות_files\ico_Filler.gif"
BORDER="0" WIDTH="3" HEIGHT="18"><IMG SRC="סלקום טקסט - משלוח
הודעות_files\ico_Filler.gif" BORDER="0" WIDTH="3" HEIGHT="18"><IMG SRC="סלקום
טקסט - משלוח הודעות_files\ico_Filler.gif" BORDER="0" WIDTH="3"
HEIGHT="18"></TD>

      <TD ALIGN="Right" VALIGN="Bottom"><A
HREF="http://192.115.11.18/webp/Cgi/PU/PU_ChangePwdForm.exe?UserName=bmidas&S
essionID=<!--write your session number here-->
&Lang=Heb"><IMG SRC="סלקום טקסט - משלוח הודעות_files\ico_UpdatePwd-Y.gif"
BORDER="0" WIDTH="79" HEIGHT="58"></A></TD>

      <TD ALIGN="Right" VALIGN="Bottom"><A
HREF="http://192.115.11.18/webp/Cgi/PU/PU_SendForm.exe?UserName=bmidas&Sessio
nID=<!--write your session number here-->
&Lang=Heb"><IMG SRC="סלקום טקסט - משלוח הודעות_files\ico_MinMain-R.gif"
BORDER="0" WIDTH="98" HEIGHT="59"></A></TD>

      <TD ALIGN="Right" VALIGN="Bottom"><A
HREF="http://www.cellcom.co.il/framemain4.html"><IMG SRC="סלקום טקסט - משלוח
הודעות_files\CellcomTextLogo.gif" BORDER="0" WIDTH="141" HEIGHT="59"></A></TD>
</TR>

<TR>
      <TD COLSPAN="5"> </TD>

      <TD BGCOLOR="#AD0000" ROWSPAN="5"> </TD>
```

```
</TR>

<TR>
      <TD ALIGN="Right" VALIGN="Top" COLSPAN="5">
      <FONT SIZE="2" COLOR="Black"><B>,        </B></FONT>
      <BR><FONT SIZE="2" COLOR="#AD0000">.דבלב טסקט סוקלס תוריש ייונמל טסקט
תועדוה חולשל לכות הז דומעמ</FONT>
      </TD>
</TR>

<TR>
      <TD ALIGN="Right" VALIGN="Top" COLSPAN="5">
      <FONT SIZE="2" COLOR="#AD0000">.סימ</FONT>
      <FONT SIZE="2" COLOR="Black"><B> 1 </B></FONT>
      <FONT SIZE="2" COLOR="#AD0000">לש הפוקתב תועדוה</FONT>
      <FONT SIZE="2" COLOR="Black"><B> 20 </B></FONT>
      <FONT SIZE="2" COLOR="#AD0000">תחילשל לבגומ רנה ,רתעידיל</FONT>
      </TD>
</TR>

<TR>
      <TD ALIGN="Right" VALIGN="Top" COLSPAN="5">
      <FONT SIZE="2" COLOR="Black"><FONT SIZE="2"
COLOR="#AD0000">.</FONT><B>06/01/2000</B></FONT>
      <FONT SIZE="2" COLOR="#AD0000">ריראתל דע</FONT>
      <FONT SIZE="2" COLOR="#AD0000">תועדוה</FONT>
      <FONT SIZE="2" COLOR="Black"><B> 19 </B></FONT>
      <FONT SIZE="2" COLOR="#AD0000">דוע רל ורתונ</FONT>
      </TD>
</TR>

<TR>
     <TD COLSPAN="5"> </TD>
</TR>

<FORM NAME="MinSendForm"
            METHOD="Post"
            ACTION="http://192.115.11.18/webp/Cgi/PU/PU_SendMessage.exe"
            onSubmit="return CheckPuSendSubmit('Heb');">

<!-- Hidden Fields -->
<INPUT TYPE="Hidden" NAME="FormType"            VALUE="MinimalMain">
<INPUT TYPE="Hidden" NAME="RecipientString">
<INPUT TYPE="Hidden" NAME="UserName"            VALUE="bmidas">
<INPUT TYPE="Hidden" NAME="SessionID"           VALUE=<!--write your session
number here-->>
<INPUT TYPE="Hidden" NAME="SentMessages"  VALUE="1">
<INPUT TYPE="Hidden" NAME="Lang"          VALUE="Heb">


<!-- Details
------------------------------------------------------------------------------>
<TR>
     <TD ALIGN="Right" VALIGN="Top" COLSPAN="5"><IMG SRC="סלקום טקסט - משלוח
הודעות_files\tit_Details-L.gif" WIDTH="472" HEIGHT="39"></TD>

     <TD ALIGN="Right" VALIGN="Top" BGCOLOR="#AD0000"><IMG SRC="- סלקום טקסט
משלוח הודעות_files\tit_Details-R.gif" WIDTH="141" HEIGHT="39"></TD>
</TR>

<TR>
     <TD ALIGN="Right" VALIGN="Top" COLSPAN="6">
```

```
      <TABLE BORDER="0" BGCOLOR="White" HEIGHT="71" WIDTH="610"
CELLPADDING="0" CELLSPACING="0">
            <TR>
                  <TD ROWSPAN="4" BGCOLOR="#FBAE07" WIDTH="2"><IMG SRC="סלקום
משלוח הודעות - טקסט_files\YellowDot.gif" WIDTH="2" HEIGHT="2"></TD>

                  <TD COLSPAN="2" WIDTH="467"><!--<IMG ALIGN="Right"
SRC="משלוח הודעות - סלקום טקסט_files\txt_EngSender.gif" WIDTH="233"
HEIGHT="15">--><IMG SRC="משלוח הודעות - סלקום טקסט_files\WhiteDot.gif"
WIDTH="2" HEIGHT="2"></TD>

                  <TD WIDTH="109" ALIGN="Left" BGCOLOR="#F4EFDD"></TD>

                  <TD WIDTH="32" ALIGN="Left" BGCOLOR="#AD0000"><IMG
SRC="משלוח הודעות - סלקום טקסט_files\RedDot.gif" WIDTH="2" HEIGHT="2"></TD>
            </TR>

            <TR>
                  <TD ALIGN="Right" BGCOLOR="White" COLSPAN="2">
                  <FONT FACE="Courier New (Hebrew)" SIZE="3">
                  <INPUT TYPE="Text" NAME="SenderName" SIZE="20"
MAXLENGTH="11">
                  </FONT>
                  </TD>

                  <TD WIDTH="109" ALIGN="Left" BGCOLOR="#F4EFDD"><IMG
SRC="משלוח הודעות - סלקום טקסט_files\lbl_SenderName.gif" WIDTH="100"
HEIGHT="17"></TD>

                  <TD WIDTH="32" ALIGN="Left" BGCOLOR="#AD0000"><IMG
SRC="משלוח הודעות - סלקום טקסט_files\RedDot.gif" WIDTH="2" HEIGHT="2"></TD>
            </TR>

            <SCRIPT>
            // Get the last sender name from the senders cookie and
            // put it in the sender name text box.
            var sPreviousSenders = GetCookie("PU_LastSenders");

            if (sPreviousSenders != null)
            {
                  asPreviousSenders = sPreviousSenders.split(";");

                  // In Netscape 4.03 and Internet Explorer 4 the method
split considers a null
                  // after the last seperator (";" in that case) as an
element in the array.
                  // In Netscape 4.05 it doesn't.
                  // The next if takes care of this situation.
                  if (asPreviousSenders.length%2)
                  {
                        nLength = asPreviousSenders.length;
                  }
                  else
                  {
                        nLength = asPreviousSenders.length + 1;
                  }

                  document.MinSendForm.SenderName.value =
asPreviousSenders[nLength - 3];
            }
            </SCRIPT>
```

```
            <TR>
                  <TD ALIGN="Right" COLSPAN="2">
                  <FONT FACE="Courier New (Hebrew)" SIZE="2">
                  <SELECT NAME="RcptAreaCode" SIZE="1">
                        <OPTION VALUE="Default" SELECTED>קידומת</OPTION>
                        <OPTION></OPTION><!-- These 2 empty options are
placeholders for the      -->
                        <OPTION></OPTION>
                        <OPTION></OPTION><!-- actual options, that are filled
later in a JS code. -->
                  </SELECT>
                  </FONT>

                  <FONT FACE="Courier New (Hebrew)" SIZE="3">
                  <INPUT TYPE="Text" NAME="RcptNumber" SIZE="9"
MAXLENGTH="6">
                  </FONT>
                  </TD>

                  <TD WIDTH="109" ALIGN="Left" BGCOLOR="#F4EFDD"><IMG
SRC="משלוח הודעות - סלקום טקסט_files\lbl_SendTo.gif" WIDTH="100"
HEIGHT="17"></TD>

                  <TD WIDTH="32" ALIGN="Left" BGCOLOR="#AD0000"><IMG
SRC="משלוח הודעות - סלקום טקסט_files\RedDot.gif" WIDTH="2" HEIGHT="2"></TD>
            </TR>

            <TR>
                  <TD WIDTH="32" ALIGN="Right" COLSPAN="2"
BGCOLOR="White"><IMG SRC="משלוח הודעות - סלקום טקסט_files\WhiteDot.gif"
WIDTH="2" HEIGHT="2"></TD>

                  <TD WIDTH="109" VALIGN="Top" ALIGN="Left"
BGCOLOR="#AD0000"><IMG SRC="משלוח הודעות - סלקום טקסט_files\lbl_Bottom.gif"
WIDTH="109" HEIGHT="12"></TD>

                  <TD WIDTH="32" ALIGN="Right" BGCOLOR="#AD0000"><IMG
SRC="משלוח הודעות - סלקום טקסט_files\RedDot.gif" WIDTH="2" HEIGHT="2"></TD>
            </TR>

            <TR>
                  <TD COLSPAN="3" BGCOLOR="#FBAE07"><IMG SRC="- סלקום טקסט
משלוח הודעות_files\YellowDot.gif" WIDTH="2" HEIGHT="2"></TD>

                  <TD WIDTH="32" COLSPAN="2" BGCOLOR="#AD0000"><IMG SRC="סלקום
משלוח הודעות - טקסט_files\RedDot.gif" WIDTH="2" HEIGHT="2"></TD>
            </TR>
      </TABLE>
      </TD>
</TR>

<!-- Message Details
------------------------------------------------------------------->
<TR>
      <TD ALIGN="Right" VALIGN="Top" COLSPAN="5"><IMG SRC="סלקום טקסט - משלוח
הודעות_files\tit_MsgDetails-L.gif" WIDTH="472" HEIGHT="29"></TD>

      <TD ALIGN="Right" VALIGN="Top" BGCOLOR="#AD0000"><IMG SRC="- סלקום טקסט
משלוח הודעות_files\tit_MsgDetails-R.gif" WIDTH="141" HEIGHT="29"></TD>
</TR>
```

```
<TR>
      <TD ALIGN="Right" VALIGN="Top" COLSPAN="6">
      <TABLE BORDER="0" BGCOLOR="White" HEIGHT="71" WIDTH="610"
CELLPADDING="0" CELLSPACING="0">
            <TR>
                  <TD ROWSPAN="4" BGCOLOR="#FBAE07" WIDTH="2"><IMG SRC="סלקום
משלוח הודעות - טקסט_files\YellowDot.gif" WIDTH="2" HEIGHT="2"></TD>

                  <TD COLSPAN="2" WIDTH="467"><IMG SRC="משלוח - סלקום טקסט
הודעות_files\WhiteDot.gif" WIDTH="2" HEIGHT="2"></TD>

                  <TD WIDTH="109" ALIGN="Left" BGCOLOR="#F4EFDD"></TD>

                  <TD WIDTH="32" ALIGN="Left" BGCOLOR="#AD0000"><IMG
SRC="משלוח הודעות - סלקום טקסט_files\RedDot.gif" WIDTH="2" HEIGHT="2"></TD>
            </TR>

            <TR>
                  <TD ALIGN="Left" VALIGN="Top"><IMG SRC="משלוח - סלקום טקסט
הודעות_files\txt_Chars.gif" WIDTH="27" HEIGHT="9">
                  <FONT FACE="Courier New (Hebrew)" SIZE="2">
                  <INPUT TYPE="Text" NAME="Counter" SIZE="4" TABINDEX="-1"
                        onChange='if (!IsNumber(this.value))
                                    {

      UpdateCounter(document.MinSendForm.CurrMsg, this);
                                    }
                        onFocus="blur();">
                  </FONT>
                  <IMG SRC="משלוח הודעות - סלקום טקסט_files\txt_Left.gif"
WIDTH="27" HEIGHT="10"></TD>

                  <TD ALIGN="Right" VALIGN="Top">
                  <!--<IMG SRC="משלוח הודעות - סלקום טקסט_files\txt_EngMsg.gif"
WIDTH="218" HEIGHT="11">-->
                  <IMG SRC="משלוח הודעות - סלקום טקסט_files\WhiteDot.gif"
WIDTH="2" HEIGHT="2">
                  </TD>

                  <TD WIDTH="109" ALIGN="Left" BGCOLOR="#F4EFDD"></TD>

                  <TD WIDTH="32" ALIGN="Left" BGCOLOR="#AD0000"><IMG
SRC="משלוח הודעות - סלקום טקסט_files\RedDot.gif" WIDTH="2" HEIGHT="2"></TD>
            </TR>

            <TR>
                  <TD ALIGN="Right" COLSPAN="2">
                  <FONT FACE="Courier New (Hebrew)" SIZE="3">
                  <INPUT TYPE="Text" NAME="CurrMsg" SIZE="44" MAXLENGTH="100"
                        onFocus="SetMessageTextFocus(true);

      UpdateAndStart(document.MinSendForm.CurrMsg,

document.MinSendForm.Counter)"
                        onBlur="SetMessageTextFocus(false);

      UpdateAndStop(document.MinSendForm.CurrMsg,

document.MinSendForm.Counter)"
                        onChange='UpdateCounter(document.MinSendForm.CurrMsg,
```

```
document.MinSendForm.Counter);

      UpdateAndStop(document.MinSendForm.CurrMsg,

document.MinSendForm.Counter)'>
                  </FONT>
                  </TD>

                  <TD WIDTH="109" ALIGN="Left" BGCOLOR="#F4EFDD"><IMG
SRC="משלוח הודעות - סלקום טקסט_files\lbl_Message.gif" WIDTH="100"
HEIGHT="17"></TD>

                  <TD WIDTH="32" ALIGN="Left" BGCOLOR="#AD0000"><IMG
SRC="משלוח הודעות - סלקום טקסט_files\RedDot.gif" WIDTH="2" HEIGHT="2"></TD>
            </TR>

            <TR>
                  <TD WIDTH="32" ALIGN="Right" COLSPAN="2"
BGCOLOR="White"><IMG SRC="משלוח הודעות - סלקום טקסט_files\WhiteDot.gif"
WIDTH="2" HEIGHT="2"></TD>

                  <TD WIDTH="109" VALIGN="Top" ALIGN="Left"
BGCOLOR="#AD0000"><IMG SRC="משלוח הודעות - סלקום טקסט_files\lbl_Bottom.gif"
WIDTH="109" HEIGHT="12"></TD>

                  <TD WIDTH="32" ALIGN="Right" BGCOLOR="#AD0000"><IMG
SRC="משלוח הודעות - סלקום טקסט_files\RedDot.gif" WIDTH="2" HEIGHT="2"></TD>
            </TR>

            <TR>
                  <TD COLSPAN="3" BGCOLOR="#FBAE07"><IMG SRC="- סלקום טקסט
משלוח הודעות_files\YellowDot.gif" WIDTH="2" HEIGHT="2"></TD>

                  <TD WIDTH="32" COLSPAN="2" BGCOLOR="#AD0000"><IMG SRC="סלקום
משלוח הודעות - טקסט_files\RedDot.gif" WIDTH="2" HEIGHT="2"></TD>
            </TR>
      </TABLE>
      </TD>
</TR>

<!-- Options
-------------------------------------------------------------------------->
<TR>
      <TD ALIGN="Right" VALIGN="Top" COLSPAN="5"><IMG SRC="משלוח - סלקום טקסט
הודעות_files\tit_Options-L.gif" WIDTH="472" HEIGHT="31"></TD>

      <TD ALIGN="Right" VALIGN="Top" BGCOLOR="#AD0000"><IMG SRC="- סלקום טקסט
משלוח הודעות_files\tit_Options-R.gif" WIDTH="141" HEIGHT="31"></TD>
</TR>

<TR>
      <TD ALIGN="Right" VALIGN="Top" COLSPAN="6">
      <TABLE BORDER="0" BGCOLOR="White" HEIGHT="35" WIDTH="610"
CELLPADDING="0" CELLSPACING="0">
            <TR>
                  <TD ROWSPAN="3" BGCOLOR="#FBAE07" WIDTH="2"><IMG SRC="סלקום
משלוח הודעות - טקסט_files\YellowDot.gif" WIDTH="2" HEIGHT="2"></TD>

                  <TD COLSPAN="2" WIDTH="467"><IMG SRC="משלוח - סלקום טקסט
הודעות_files\WhiteDot.gif" WIDTH="2" HEIGHT="2"></TD>
```

```
                    <TD WIDTH="109" ALIGN="Left" BGCOLOR="#F4EFDD"></TD>

                    <TD WIDTH="32" ALIGN="Left" BGCOLOR="#AD0000"><IMG
SRC="משלוח הודעות - סלקום טקסט_files\RedDot.gif" WIDTH="2" HEIGHT="2"></TD>
              </TR>

              <TR>
                    <TD ALIGN="Right" COLSPAN="2">
                    <FONT FACE="Courier New (Hebrew)" SIZE="2">
                    <SELECT NAME="Urgency" SIZE="1">
                          <OPTION>________</OPTION>
                          <OPTION></OPTION><!-- These 3 empty options are
placeholders for the      ------>
                          <OPTION></OPTION><!-- actual options, that are filled
later in a JS code. ------>
                    </SELECT>
                    </FONT>
                    </TD>

                    <TD WIDTH="109" ALIGN="Left" BGCOLOR="#F4EFDD"><IMG
SRC="משלוח הודעות - סלקום טקסט_files\lbl_Urgency.gif" WIDTH="100"
HEIGHT="17"></TD>

                    <TD WIDTH="32" ALIGN="Left" BGCOLOR="#AD0000"><IMG
SRC="משלוח הודעות - סלקום טקסט_files\RedDot.gif" WIDTH="2" HEIGHT="2"></TD>
              </TR>

              <TR>
                    <TD WIDTH="32" ALIGN="Right" COLSPAN="2"
BGCOLOR="White"><IMG SRC="משלוח הודעות - סלקום טקסט_files\WhiteDot.gif"
WIDTH="2" HEIGHT="2"></TD>

                    <TD WIDTH="109" VALIGN="Top" ALIGN="Left"
BGCOLOR="#AD0000"><IMG SRC="משלוח הודעות - סלקום טקסט_files\lbl_Bottom.gif"
WIDTH="109" HEIGHT="12"></TD>

                    <TD WIDTH="32" ALIGN="Right" BGCOLOR="#AD0000"><IMG
SRC="משלוח הודעות - סלקום טקסט_files\RedDot.gif" WIDTH="2" HEIGHT="2"></TD>
              </TR>

              <TR>
                    <TD COLSPAN="3" BGCOLOR="#FBAE07"><IMG SRC="- סלקום טקסט
משלוח הודעות_files\YellowDot.gif" WIDTH="2" HEIGHT="2"></TD>

                    <TD WIDTH="32" COLSPAN="2" BGCOLOR="#AD0000"><IMG SRC="סלקום
משלוח הודעות - טקסט_files\RedDot.gif" WIDTH="2" HEIGHT="2"></TD>
              </TR>
        </TABLE>
        </TD>
</TR>

<!-- Send
-------------------------------------------------------------------------
->
<TR>
      <TD ALIGN="Left" VALIGN="Center" COLSPAN="5"><INPUT TYPE="IMAGE"
SRC="משלוח הודעות - סלקום טקסט_files\btn_SendMsg.gif" BORDER="0" WIDTH="104"
HEIGHT="34"></TD>

      <TD ALIGN="Right" VALIGN="Top"><IMG SRC="משלוח - סלקום טקסט
הודעות_files\sbr_Bottom.gif" BORDER="0" WIDTH="141" HEIGHT="58"></TD>
```

```
</TR>

</FORM>
</TABLE>

<SCRIPT>
var nValidTime = 30;
var dToday = new Date();
var dExpires = new Date();
dExpires.setTime(dToday.getTime() + 1000*60*nValidTime);
SetCookie("PU_CellcomTextLogin", "bmidas"+";"+"bmidas", dExpires);

SetAreaCodesInSelectBox(document.MinSendForm.RcptAreaCode, true);
SetUrgenciesInSelectBox(document.MinSendForm.Urgency, false);
fillForm();
</SCRIPT>

<!--</DIV>
</BODY>
</HTML>-->

<!--<A HREF="aaa">
<IMG  SRC="משלוח הודעות - סלקום טקסט_files\Footer.gif"
            BORDER="0"
            ISMAP>
</A>-->

<IMG  SRC="משלוח הודעות - סלקום טקסט_files\Footer.gif"
            HEIGHT="47"
            WIDTH="619"
            BORDER="0"
            USEMAP="#FooterMap">

<MAP NAME="FooterMap">

<AREA SHAPE="Rect"
            COORDS="320,0,375,24"
            TARGET="Cellcom"
            HREF="http://www.cellcom.co.il"
            TITLE="Cellcom home page">

<AREA SHAPE="Rect"
            COORDS="64,27,108,47"
            TARGET="Box"
            HREF="http://www.box.co.il"
            TITLE="Box home page">

<AREA SHAPE="Rect"
            COORDS="552,27,619,47"
            TARGET="Netology"
            HREF="http://www.netology-sms.com"
            TITLE="Netology home page">

</MAP>
</DIV>
</BODY>
</HTML>
```

Module 6 - Example of enriched data sent from server (306 in fig 3a) to personal agent (340) coupled to a query. If personal agent (340) analyzes the client's database (320) through the DMS (330) according to the query coupled

```
to this HTML page and tags it relevant, this HTML page is shown to the client
(Transaction 311)
>HTML>
>HEAD>
>META NAME="GENERATOR" Content="Microsoft Visual Studio 6.0">
>TITLE></TITLE>
/>HEAD>
>BODY>

>P align=center><FONT color=fuchsia><STRONG/>יתרה שלילית'<STRONG></FONT>  </P>

/>BODY>
/>HTML>
```

Module 7 - This module contains exemplary Database commands which are not implemented by the other protocols. This is an example of accessing a text database.

a. This function accesses an html downloaded from a service provider and saves it in a text file.

```
'*************************************************************
'* DESCRIPTION : import info from heshbonall.html            *
'*               and update table tnuot                      *
'* order of fields in table:Date - תאריך                     *
'*                          strHeshbon - מספר חשבון          *
'*                          dtUpdated  - תאריך ייצור דף      *
'*                          asmachta   - אסמכתא              *
'*                          nose   -מקור                     *
'*                          tnua - תנועה                     *
'*                          itra - יתרה                      *
'*                          nowdate -תאריך תנועה             *
'*************************************************************

Sub ImporttnuotHTML()
    Dim strNewLine As String        'input line from html file
    Dim strYearTemp   As String     'year of update
    Dim strHeshbon As String        'number of account
    Dim a As Integer                'checks if new account (a=1)
    Dim dtUpdated As Date           'updating date
    Dim fileNumber As Integer       'number of input file
    Dim dtUpdateTizmun As Date      'gets date of importing for tizmun table
    Dim intTimes                    'number of accounts
    Dim fileNumberOpen As Boolean       'is fileNumber open
    Dim intFileNumberDB As Integer      'number of db file
    Dim intFileNumberTemp As Integer    'number of temp file
    Dim blnFileDbOpen As Boolean        'is db file open
    Dim blnFileTempOpen As Boolean          'is temp file open
    Dim arrResult(100, 7)  As String    'array will hold result
    Dim line(7) As String
    Dim rows As Integer                 'row number
    Dim columns As Integer              'columns number
    rows = 0
    columns = 0


    blnFileDbOpen = False
    blnFileTempOpen = False


    fileInputOpen = False
    fileOutputOpen = False
```

```
        fileNumberOpen = False

        intTimes = 0            'opening html file to read data
        fileNumber = FreeFile()
        Open AppPath & "\heshbonall.htm" For Input As #fileNumber
        fileNumberOpen = True

        intFileNumberTemp = FreeFile()
        Open AppPath & "\temp.txt" For Output As #intFileNumberTemp
        blnFileTempOpen = True

        Line Input #fileNumber, strNewLine
        strNewLine = TernUpperToLower(strNewLine)
        Do While Not (EOF(fileNumber))
            If intTimes > 0 Then
                    If Leumi = True Then
                        itrot(countHeshbon, 0) = "בנק לאומי"
                    Else
                        itrot(countHeshbon, 0) = "בנק הפועלים"
                    End If
                    itrot(countHeshbon, 1) = strHeshbonCopy
                    itrot(countHeshbon, 2) = itra
                    itrot(countHeshbon, 3) = dtUpdatedCopy
                    countHeshbon = countHeshbon + 1
            End If
            Do Until Mid(strNewLine, 1, 26) = "<font color=white>|</font>" Or
EOF(fileNumber) Or Mid(strNewLine, 1, 26) = "<font color=white>a</font>"
                    Line Input #fileNumber, strNewLine
                    strNewLine = TernUpperToLower(strNewLine)
            Loop
            If EOF(fileNumber) Then
                    Exit Do
            End If
            Do Until (strNewLine Like "*תאריך*") Or EOF(fileNumber)
                    Line Input #fileNumber, strNewLine
                    'strNewLine = TernUpperToLower(strNewLine)
            Loop
            If EOF(fileNumber) Then
                    Exit Do
            End If
            Line Input #fileNumber, strNewLine

            dtUpdated = Mid(strNewLine, 77, 8)
            dtUpdatedCopy = dtUpdated

           strYearTemp = Mid(strNewLine, 83, 2)
           Line Input #fileNumber, strNewLine

           strHeshbon = Mid(strNewLine, 71, 9)
           strHeshbonCopy = strHeshbon
           intTimes = intTimes + 1
           'dtUpdateTizmun = Mid(strNewLine, 98, 8)
           Line Input #fileNumber, strNewLine
           Line Input #fileNumber, strNewLine
           Line Input #fileNumber, strNewLine
           strNewLine = TernUpperToLower(strNewLine)
           Do While Mid(strNewLine, 1, 26) = "<font color=white>|</font>"
                If (Mid(strNewLine, 80, 1) = "/") And (Mid(strNewLine, 68, 8) <>
"בתוקף עד") Then

                    'writing itra יתרה
                    If Mid(strNewLine, 27, 13) = "              " Then
```

```
                    itra = " "
                Else
                    itra = Mid(strNewLine, 29, 13)
                    If Mid(strNewLine, 28, 1) = "ת" Then
                        itra = -1 * itra
                    End If
                End If
                'writing value תנועה
                tnua = IIf(Mid(strNewLine, 54, 4) <> "    ",
Mid(strNewLine, 50, 16), Mid(strNewLine, 60, 16))
                If Mid(strNewLine, 54, 4) = "    " Then tnua = -1 * tnua

                'writing makor נושא
                nose = Trim(Mid(strNewLine, 93, 14))
                nose1 = ""
                For I = 1          se)
                    let            e, I, 1)
                    I              <> 34 Then
                                = nose1 & lett

                    im(nose1)

                    ng asmachta אסמכתא
                  achta = Mid(strNewLine, 84, 8)

                'writing date תאריך
                nowdate = Mid(strNewLine, 78, 2) & "/" & Mid(strNewLine,
81, 2) & "   " strYearTemp
                todayDate = Format(Date, "dd/mm/yy")
                pageDate = Format(dtUpdated, "dd/mm/yy")

                Write #intFileNumberTemp, todayDate; strHeshbon;
pageDate; asmachta; nose; tnua; itra; nowdate
                arrResult(rows, 0) = strHeshbon
                arrResult(rows, 1) = pageDate
                arrResult(rows, 2) = asmachta
                arrResult(rows, 3) = nose
                arrResult(rows, 4) = tnua
                arrResult(rows, 5) = itra
                arrResult(rows, 6) = nowdate
                rows = rows + 1
            End If
            Line Input #fileNumber, strNewLine
            strNewLine = TernUpperToLower(strNewLine)
        Loop
    Loop
    intFileNumberDB = FreeFile()
    Open AppPath & "\tnuotDb.txt" For Input As #intFileNumberDB
     blnFileDbOpen = True
      rows = rows - 1
    Do Until EOF(intFileNumberDB)
        Input #intFileNumberDB, X
        If EOF(intFileNumberDB) Or X = "" Then
            Exit Do
        End If
        Input #intFileNumberDB, line(0), line(1), line(2), line(3),
line(4), line(5), line(6)

         If (IsIN(arrResult(), line(), 7, rows) = False) Then
            Write #intFileNumberTemp, X; line(0); line(1); line(2);
line(3); line(4); line(5); line(6)
```

```
            Else
               blnNeedImportOld = False
            End If
        Loop
        Close #fileNumber
        fileNumberOpen = False
        Close #intFileNumberDB
        blnFileDbOpen = False
        Close #intFileNumberTemp
        blnFileTempOpen = False
        FileCopy AppPath & "\temp.txt", AppPath & "\tnuotDb.txt"
        Kill AppPath & "\temp.txt"
        UpdateTizmun "heshbonall.htm", todayDate
        Exit Sub
ImporttnuotHTMLErrHandler:

    strOriginal = AppPath & "\heshbonall.htm"
    strNow = Date
    strNow = Mid(strNow, 1, 2) & Mid(strNow, 4, 2) & Mid(strNow, 7, 2)
    strNew = AppPath & "\bank\tnuotTable" & strNow & ".htm"
    If fileNumberOpen = True Then
        Close #intFileNumber
    End If
    If blnFileDbOpen = True Then
        Close #intFileNumberDB
    End If
    If blnFileTempOpen = True Then
        Close #intFileNumberTemp
    End If

    'HandError strOriginal, strNew
    OnError "ImporttnuotHTML", Err, Error$, Date


End Sub
```

b. This fuction accesses a text file and returns the number of fields in that file.

```
Function GetNumOfFields(ByVal tableName As String) As Integer
Select Case (tableName)
    Case "tnuotDb":
        GetNumOfFields = 8
    Case "pikdonotDb":
        GetNumOfFields = 9
    Case "visaDb":
        GetNumOfFields = 9
    Case "matzavhmti":
        GetNumOfFields = 9
    Case "gemelDb":
         GetNumOfFields = 12
    Case "hisahonDb":
          GetNumOfFields = 11
    Case "niarotDb":
          GetNumOfFields = 10
    Case "poalimgemelDB":
        GetNumOfFields = 10
    Case "poalimHisahonDb":
        GetNumOfFields = 8
    Case "poalimpikdonotDb":
        GetNumOfFields = 8
    Case "poalimvisaDb":
        GetNumOfFields = 6
    Case "poalimniarotDB":
```

```
        GetNumOfFields = 8
    Case "poalimitrotDB":
        GetNumOfFields = 6
    Case "poalimtnuotDB":
        GetNumOfFields = 7
    End Select
End Function
```

c. This function accesses a text file and returns the structure of the file.

```
Public Function AskQuery1(ByVal tableName As String) As udRecord()
    Dim intFileNumber As Integer 'file number
    Dim temp() As String
    Dim intLoopCounter
    Dim intRowsCounter
    Dim size As Integer
    Dim arrResult1()  As udRecord

    ReDim temp(0)
    ReDim arrResult1(0)
    intLoopCounter = 0
    intRowsCounter = 0
    size = GetNumOfFields(tableName) - 1
    intFileNumber = FreeFile()
    c = AppPath
    Open AppPath & tableName & ".txt" For Input As #intFileNumber
    If EOF(intFileNumber) Then
        AskQuery1 = Null
        Exit Function
    End If

    Do Until EOF(intFileNumber)
        For intLoopCounter = 0 To size
            Input #intFileNumber, temp(intLoopCounter)
            temp(intLoopCounter) = Trim(temp(intLoopCounter))
            ReDim Preserve temp(UBound(temp) + 1)
        Next intLoopCounter
        'ReDim Preserve arrResult1(UBound(arrResult1) + 1)
        arrResult1(intRowsCounter).row = temp
        ReDim Preserve arrResult1(UBound(arrResult1) + 1)
        intRowsCounter = intRowsCounter + 1
    Loop
     ReDim Preserve arrResult1(UBound(arrResult1) - 1)
'    End Select
    Close #intFileNumber
    AskQuery1 = arrResult1
   ' arrResult = SelectWhere(arrResult1(), 7, ">", "05/12/99", 3, "=",
"4032")
  '  a = GetMaxMin(arrResult1(), 7, "min")
     'arrResult = SelectDifferent(arrResult1(), 2)
     'arrResult = Miyun1(arrResult1(), 7)
    'MsgBox ("" & a)
End Function
```

d. This function receives a recordset and returns the recordset after being grouped.

```
Function SelectDifferent(array1() As udRecord, ByVal fieldNumber As Integer)
As udRecord()
    Dim values() As udRecord
    Dim intNumOfRecords As Integer
    Dim binFound As Boolean
    Dim I As Integer
```

```
    Dim n As Integer
    Dim k As Integer

    ReDim values(0)
    intNumOfRecords = UBound(array1) - 1
    values(0) = array1(0)
    For I = 1 To intNumOfRecords
        n = UBound(values())
        blnFound = False
        For k = 0 To n
            If array1(I).row(fieldNumber) = values(k).row(fieldNumber) Then
                blnFound = True
            End If
        Next k
        If blnFound = False Then
            ReDim Preserve values(UBound(values) + 1)
            values(UBound(values)) = array1(I)
        End If
    Next I
    SelectDifferent = values()
End Function
```

e. This function receives a recordset and returns the recordset after being sorted.

```
Function Miyun1(array1() As udRecord, fieldNumber As Integer, ByVal
directionSort As String) As udRecord()
    Dim arraya() As udRecord
    Dim arrayb() As udRecord
    Dim arrayc() As udRecord
    Dim q As Integer
    Dim I As Integer
    Dim f As Integer
    Dim k As Integer

    ReDim arraya(0)
    ReDim arrayb(0)
    sizeArray = UBound(array1())
    If sizeArray = 0 Then
        Miyun1 = array1()
        Exit Function
    Else
        q = ((1 + sizeArray) / 2)
        If q * 2 > sizeArray + 1 Then
            q = q - 1
        End If
        For I = 0 To q - 1
           ReDim Preserve arraya(I)
           arraya(I) = array1(I)
        Next I
        f = 0
        For k = q To sizeArray
            ReDim Preserve arrayb(f)
            arrayb(f) = array1(k)
            f = f + 1
        Next k
        arraya = Miyun1(arraya(), fieldNumber, directionSort)
        arrayb = Miyun1(arrayb(), fieldNumber, directionSort)
        arrayc = Miyun2(arraya(), arrayb(), fieldNumber, directionSort)
        Miyun1 = arrayc
    End If
End Function
```

f. This function receives a name of a table, field to sort by, sort order, and criteria. And returns a recordset.

```
'*********************************************************
'*DESCRIPTION: this function quries a table
'*INPUT:       tableName - name of table (string)
'*             fieldSort - number of field to sort by
'*             directionSort - direction of sorting:
'*             "HtoL"
'*             "LtoH"
'*             ParamArray  arrayof conditions:
'*             1,"<",30,3,"=","01/06/00",1,"=","97584"
'*********************************************************
Function QuerTable(ByVal tableName As String, ByVal fieldSort As Integer,
ByVal directionSort As String, ParamArray argArray() As Variant)
    Dim res1() As udRecord
    Dim res2() As udRecord
    Dim res3() As udRecord
    Dim argArray1() As Variant


    X = UBound(argArray)
    ReDim argArray1(0)
    For I = 0 To X
        ReDim Preserve argArray1(I)
        argArray1(I) = argArray(I)
    Next I

    res1() = AskQuery1(tableName)
    If UBound(argArray()) >= 2 Then
        res2() = SelectWhere(res1(), argArray1())
        If fieldSort <> 100 Then
            res3() = Miyun1(res2(), fieldSort, directionSort)
            arrResult = res3
        Else
            arrResult = res2
        End If
    Else
         If fieldSort <> 100 Then
            res3() = Miyun1(res1(), fieldSort, directionSort)
            arrResult = res3
        Else
            arrResult = res1
        End If
    End If
End Function
```

g. This function receives a recordset, field to check, Min or Max. And returns the ID of the record containing the Max or Min.

```
Function GetMaxMin(array1() As udRecord, ByVal fieldNumber As Integer, ByVal
maxOrMin As String) As String
    Dim intSize As Integer
    Dim strMinMax As String

    intSize = UBound(array1())
    strMinMax = array1(0).row(fieldNumber)
    Select Case (maxOrMin)
    Case ("min"):
        For I = 0 To intSize
            If array1(I).row(fieldNumber) < strMinMax Then
                strMinMax = array1(I).row(fieldNumber)
            End If
```

```
        Next I
    Case ("max"):
        For I = 0 To intSize
            If array1(I).row(fieldNumber) > strMinMax Then
                strMinMax = array1(I).row(fieldNumber)
            End If
        Next I
    End Select
    GetMaxMin = strMinMax
End Function
```

h. This function checks, using the above functions, if the users bank account is in overdraft. This example illustrates the Personal Agent checking the personal financial status of its user.

```
Public Function ChekRedFlagsLomi()

 y() = AskQuery1("tnuotDb")
  g = SelectDifferent(y, 1)
   z = UBound(g)

  For w = 0 To z


   a = QuerTable("tnuotDb", 7, "HtoL", 1, "=", g(w).row(1))
         s = GetSize()
         l = GetNumOfFields("tnuotDb") - 1
           X = ""
     For I = 0 To s
      X = ""
       position = X & " " & ReturnValue(I, 6)
        position = Trim(position)
          strOver = Mid(position, 1, 1)

      If strOver = "-" Then
         For k = 0 To l
           X = X & " " & ReturnValue(I, k)
         Next k
         frmChekRedFlags.List1.AddItem (X)
      End If
     Next I

  Next w

frmChekRedFlags.List1.ListIndex = 0
 strMessageText = Trim(frmChekRedFlags.List1.Text)
              MsgBox strMessageText

End Function
```

What is claimed is:

1. A system for directing a blind solicitation to a pre-definable, anonymous potential customer client computer, comprising:
   i. a supplier and a query aggregating server, said query aggregating server communicatively connected to a network, said query aggregating server including programming for forwarding a query upon receiving a recognized request, said query comprising an offer from said supplier and a definable characteristics profile for identifying a potential customer; and
   ii. a customer's client computer, communicatively connectable to said network, said customer's client computer further comprising a personal agent for requesting and receiving a query from said query aggregating server via said network, said customer's client computer further comprising personal databases accessible to said personal agent, said personal agent further including programming for executing said query, scanning data in said personal databases and determining the relevance to said customer of said offer contained in said query, and for notifying said customer of said offer if said determination for relevance is positive,

   whereby no identifying information about said potential customer is returned or accessible to said supplier.

2. The system for directing a blind solicitation to a pre-definable, anonymous potential customer client computer device, according to claim 1, wherein said said query aggregating server is part of said supplier's network.

3. The system for directing a blind solicitation to a pre-definable, anonymous potential customer client computer device, according to claim 2, wherein said query is downloaded to said client computer as part of the code of said supplier's Webpage.

4. The system for directing a blind solicitation to a pre-definable, anonymous potential customer client computer device, according to claim 1, where in said device is a communication device.

5. The system of claim 2, wherein said communication device is selected from the group consisting of: Personal Data Assistants (PDA),Wireless Application Protocol (WAP), telephones, cellular phone, e-mail, laptops, personal computers.

6. The system for directing a blind solicitation to a pre-definable, anonymous potential customer client computer device, according to claim 1, wherein said personal agent is activated automatically.

7. The system for directing a blind solicitation to a pre-definable, anonymous potential customer client computer device, according to claim 1, wherein said personal agent is activated according to a time period, programmed by the user.

8. The system for directing a blind solicitation to a pre-definable, anonymous potential customer client computer device, according to claim 1, wherein the personal agent contains a user preference checklist enabling the client to choose from which suppliers he wants to get offers.

9. The system for directing a blind solicitation to a pre-definable, anonymous potential customer client computer device, according to claim 1, wherein the personal agent contains a user preference checklist enabling the user to choose the subject matter about which he wants to get offers.

10. The system for directing a blind solicitation of claim 1 wherein, said query could be updated by said supplier.

11. The system for directing a blind solicitation of claim 1 wherein, said query could be updated by said query-aggregating server according to said supplier needs.

12. The system for directing a blind solicitation of claim 1 wherein, said client can activate an offer responding machine.

13. The system for directing a blind solicitation in accordance with claim 12, wherein said user can choose to which suppliers said offer responding machine should reply.

14. The system for directing a blind solicitation in accordance with claim 12, wherein said offer responding machine may be set to automatically reply according to the subject matter of said offer.

15. A method for directing a blind offer to a pre-definable, anonymous potential customer client device, in a system including a supplier and a query aggregating service provider, said supplier desiring to present an offer to said pre-definable, anonymous potential customer's client device, comprising the steps of:

(a) selection by said supplier of a definable characteristics profile for identifying a potential customer;

(b) forming a query from said profile and said offer;

(c) receiving a request from said customer's client device to said query aggregating service provider for downloading a query;

(d) forwarding said query, from said query aggregating service provider to said customer's client device;

(e) determining the relevance to said customer of said offer contained in said query by scanning a personal database in said client device, upon receiving said query from said query aggregating service provider, said scanning being conducted according to said definable characteristics profile and notifying said customer of said offer if said determination for relevance is positive.

16. A method in accordance with claim 15 for directing a blind offer to a pre-definable, anonymous potential customer client device, in a system including a supplier and a query aggregating service provider, said supplier desiring to present an offer to said pre-definable, anonymous potential customer's client device, further comprising the step of enriching said offer with personal information obtained from said personal database.

Fig 1a
Prior Art

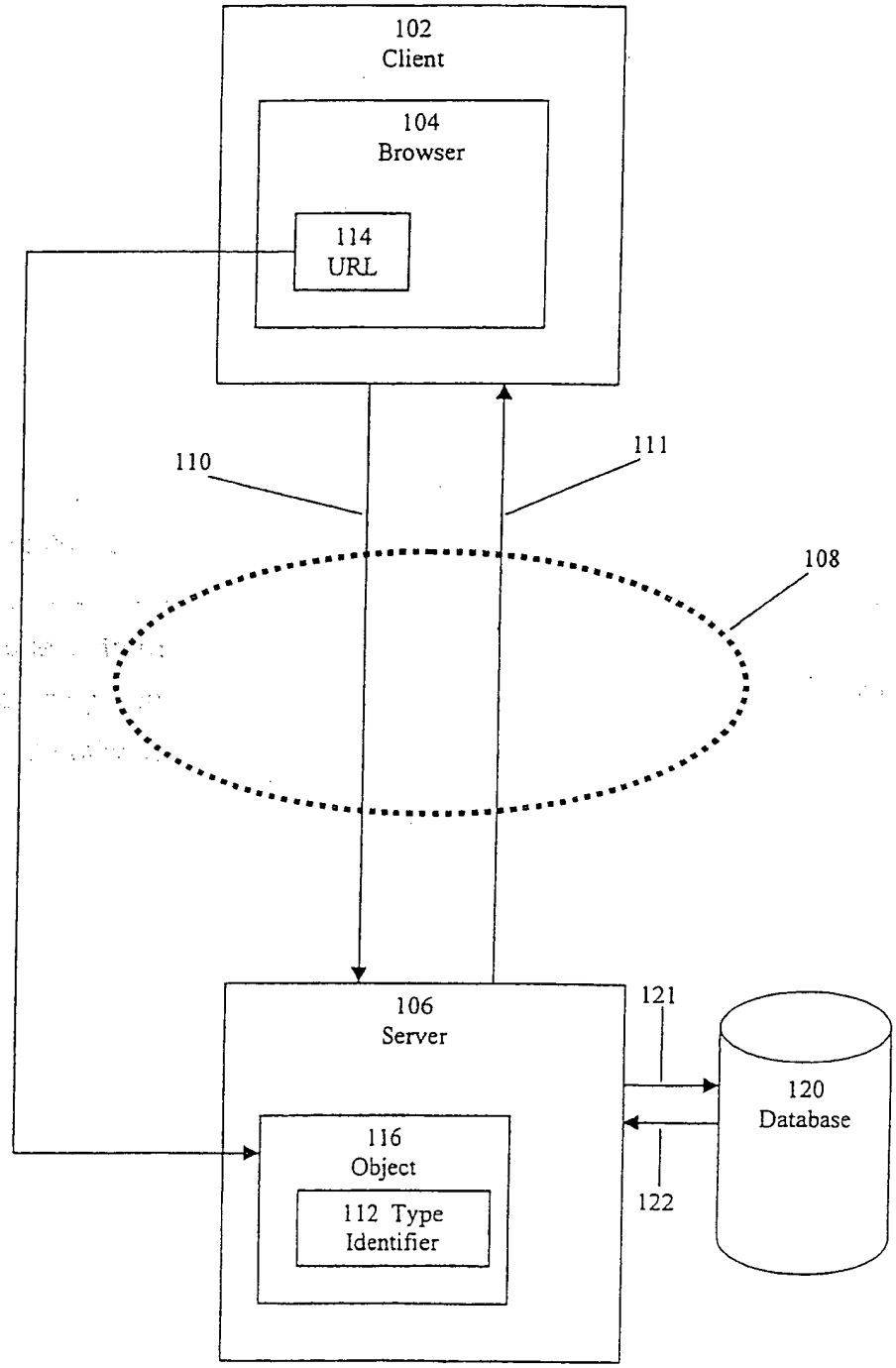


Fig 1b
Prior Art

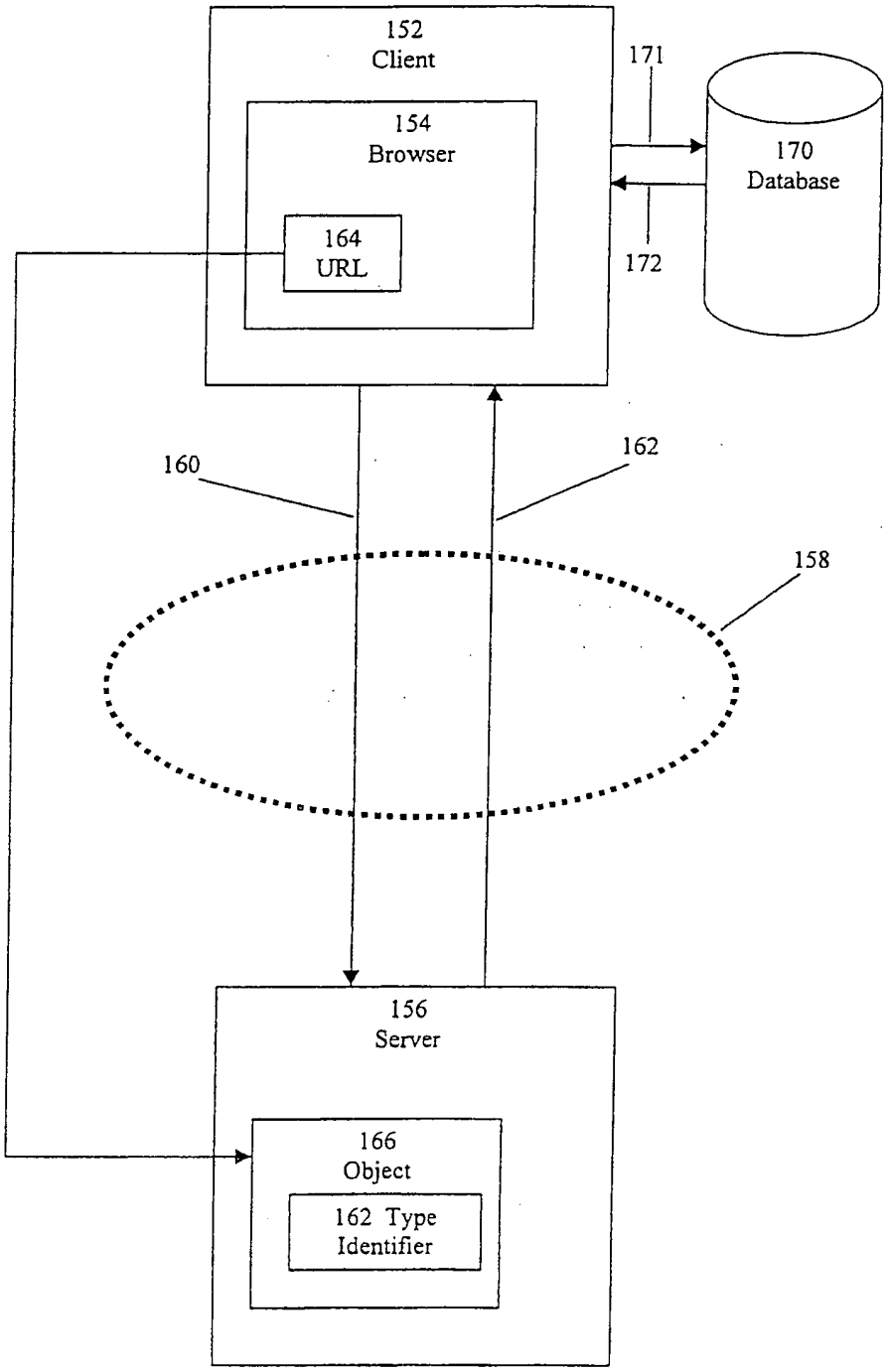


Fig 2a

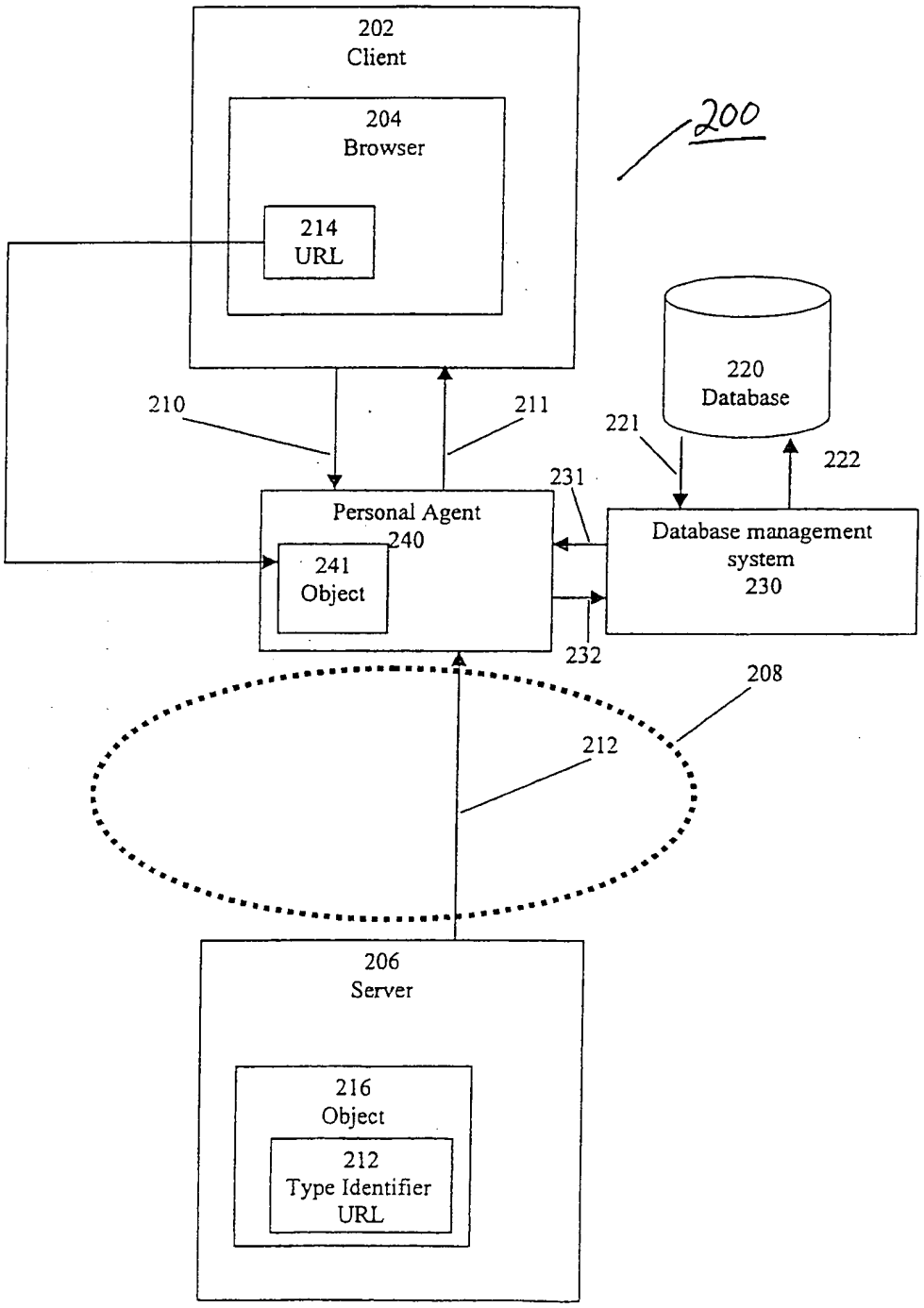

202
Client
204
Browser
214
URL
200
220
Database
210
211
221
222
231
Personal Agent
240
241
Object
Database management
system
230
232
208
212
206
Server
216
Object
212
Type Identifier
URL

Fig 2b

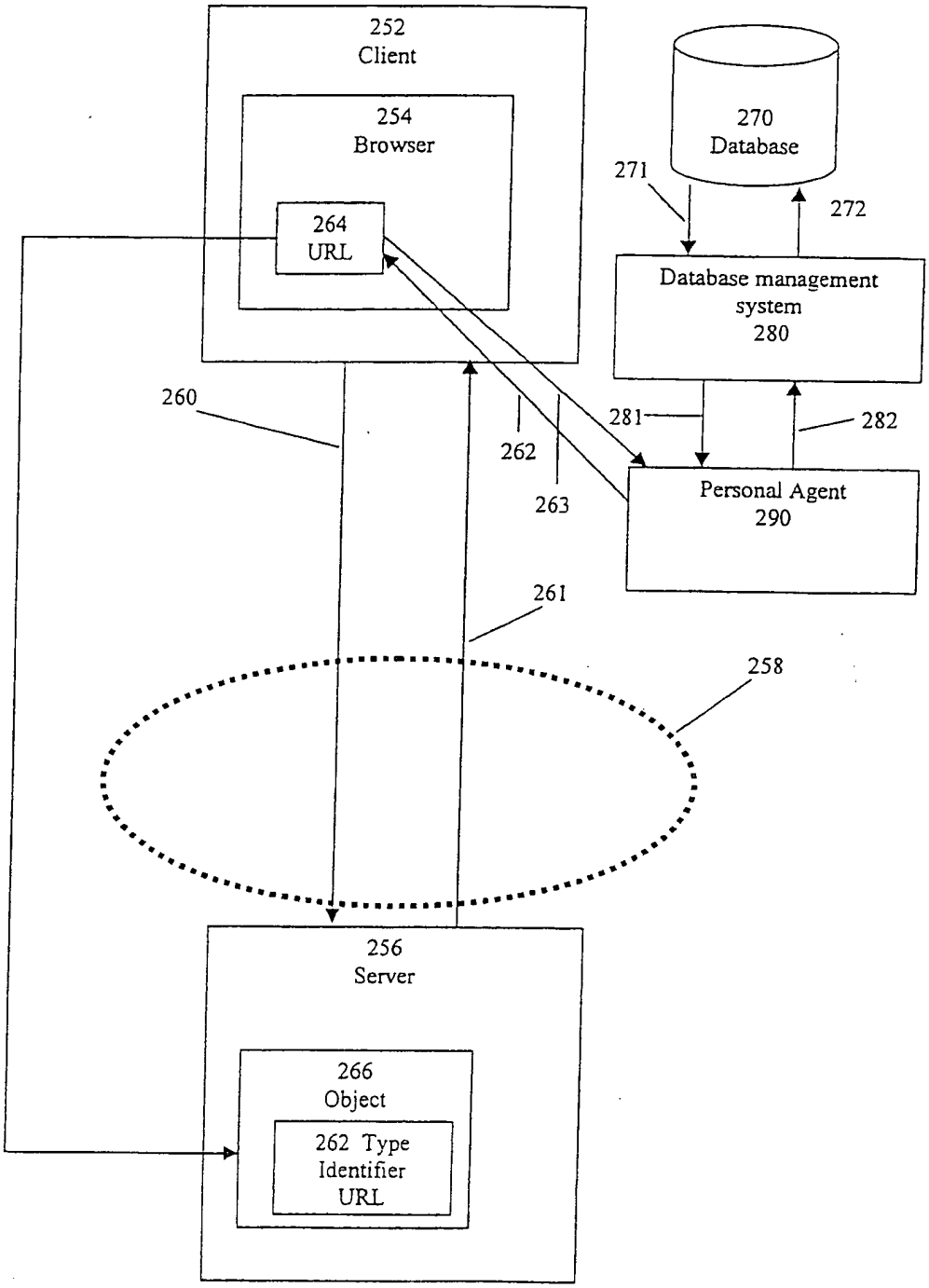

252
Client
254
Browser
264
URL
270
Database
271
272
Database management
system
280
260
281
282
262
263
Personal Agent
290
261
258
256
Server
266
Object
262 Type
Identifier
URL

Fig 3a

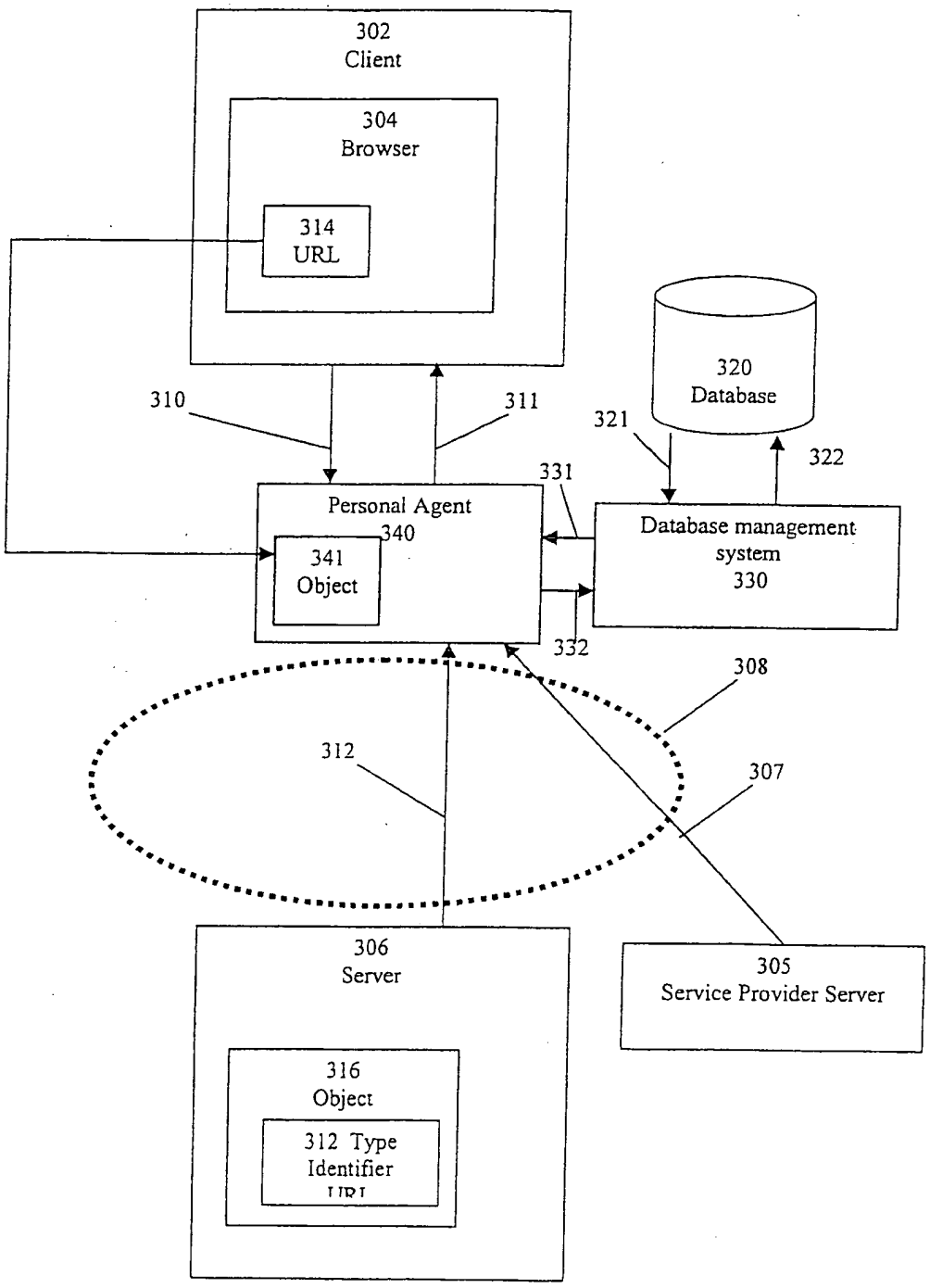

302
Client
304
Browser
314
URL
320
Database
310
311
321
322
331
Personal Agent
340
341
Object
Database management
system
330
332
308
312
307
306
Server
316
Object
312 Type
Identifier
URI
305
Service Provider Server

Fig 3b

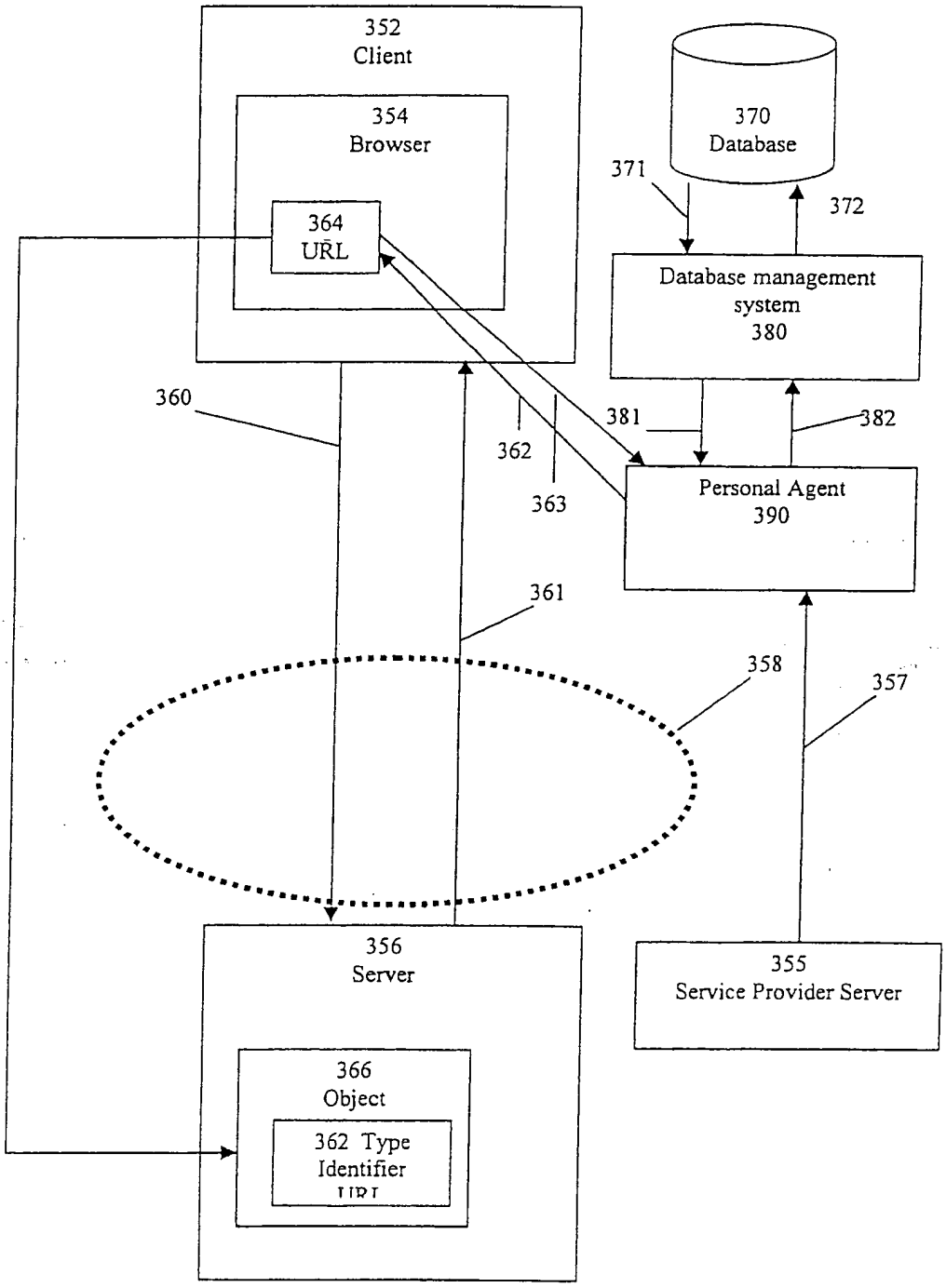

352
Client
354
Browser
364
URL
370
Database
371
372
Database management system
380
360
381
382
362
363
Personal Agent
390
361
358
357
356
Server
355
Service Provider Server
366
Object
362 Type
Identifier
URL

Fig 4a

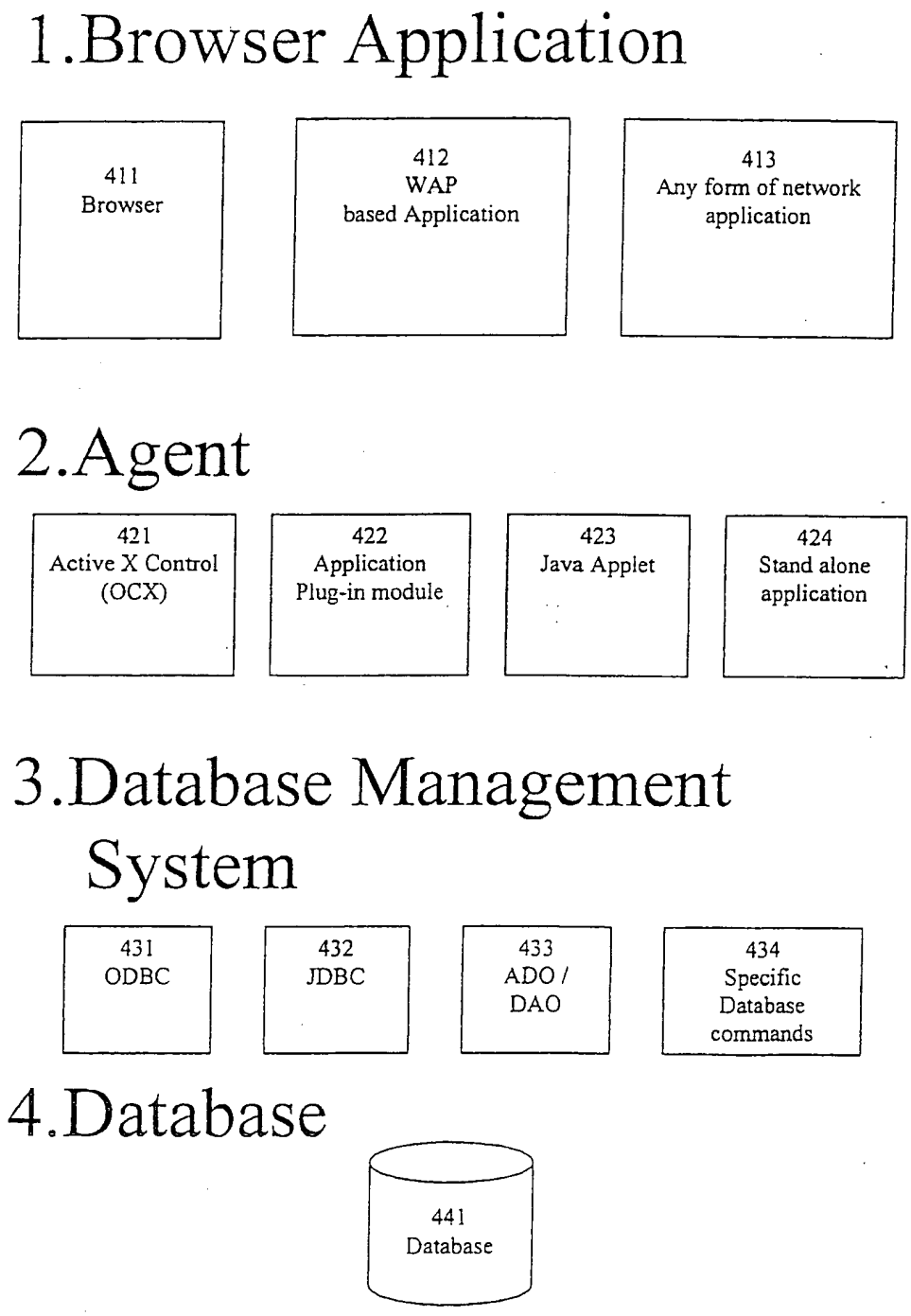
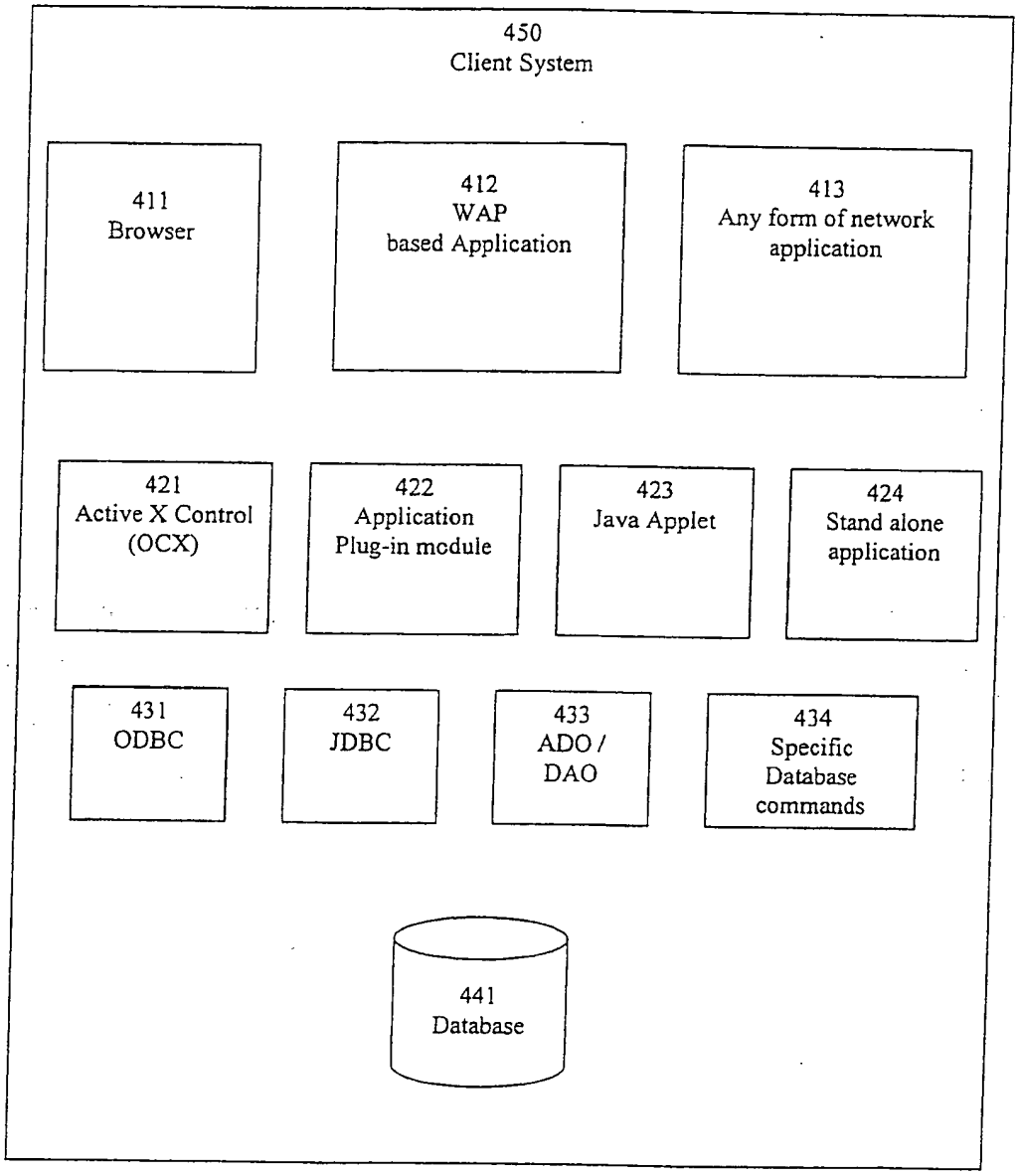

1.Browser Application
411
Browser
412
WAP
based Application
413
Any form of network
application
2.Agent
421
Active X Control
(OCX)
422
Application
Plug-in module
423
Java Applet
424
Stand alone
application
3.Database Management
System
431
ODBC
432
JDBC
433
ADO /
DAO
434
Specific
Database
commands
4.Database
441
Database

Fig 4b


450
Client System
411
Browser
412
WAP
based Application
413
Any form of network
application
421
Active X Control
(OCX)
422
Application
Plug-in module
423
Java Applet
424
Stand alone
application
431
ODBC
432
JDBC
433
ADO /
DAO
434
Specific
Database
commands
441
Database

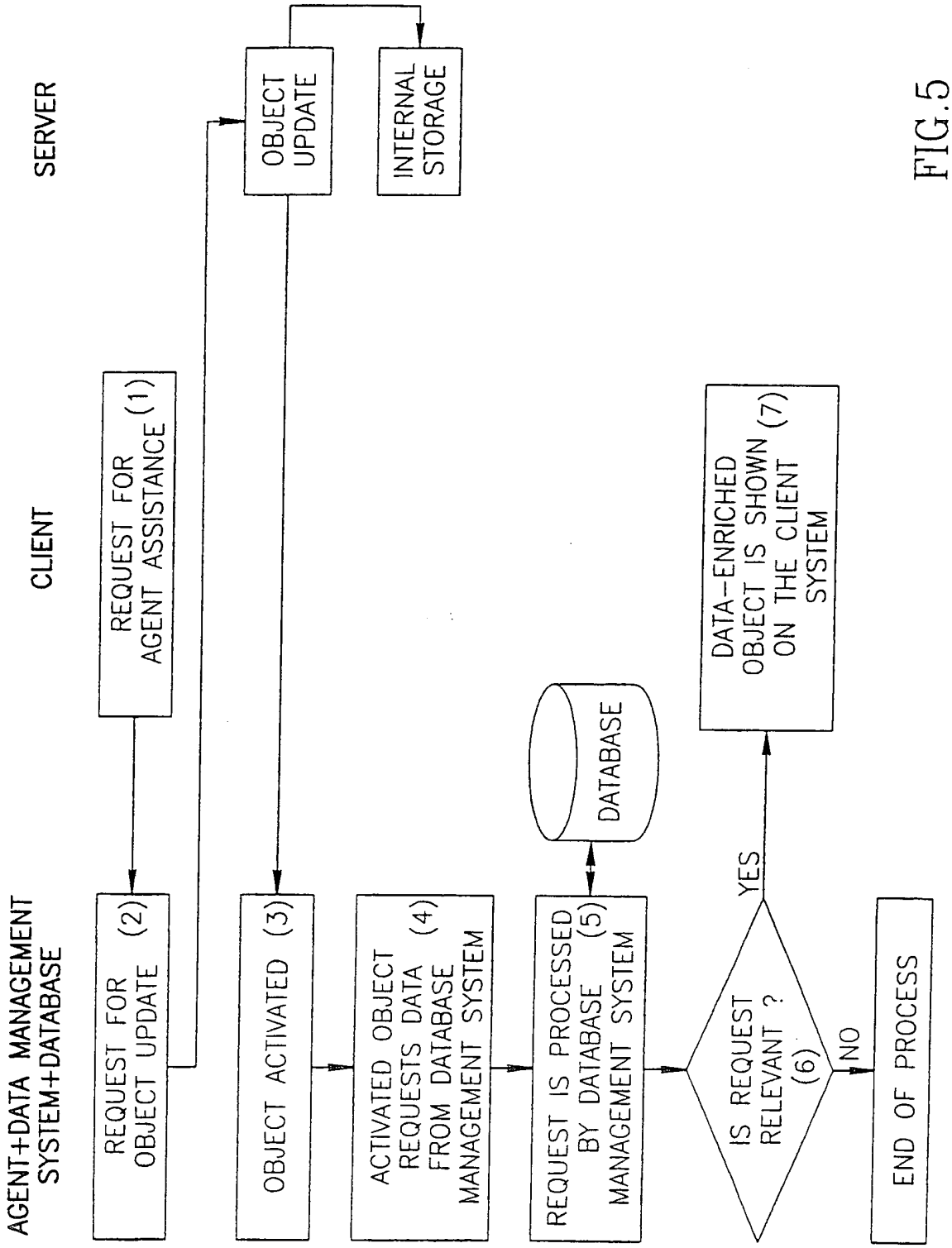

AGENT+DATA MANAGEMENT SYSTEM+DATABASE
CLIENT
SERVER
REQUEST FOR AGENT ASSISTANCE (1)
REQUEST FOR OBJECT UPDATE (2)
OBJECT UPDATE
INTERNAL STORAGE
OBJECT ACTIVATED (3)
ACTIVATED OBJECT REQUESTS DATA FROM DATABASE MANAGEMENT SYSTEM (4)
DATABASE
REQUEST IS PROCESSED BY DATABASE MANAGEMENT SYSTEM (5)
IS REQUEST RELEVANT ? (6)
YES
NO
DATA-ENRICHED OBJECT IS SHOWN ON THE CLIENT SYSTEM (7)
END OF PROCESS

FIG.5

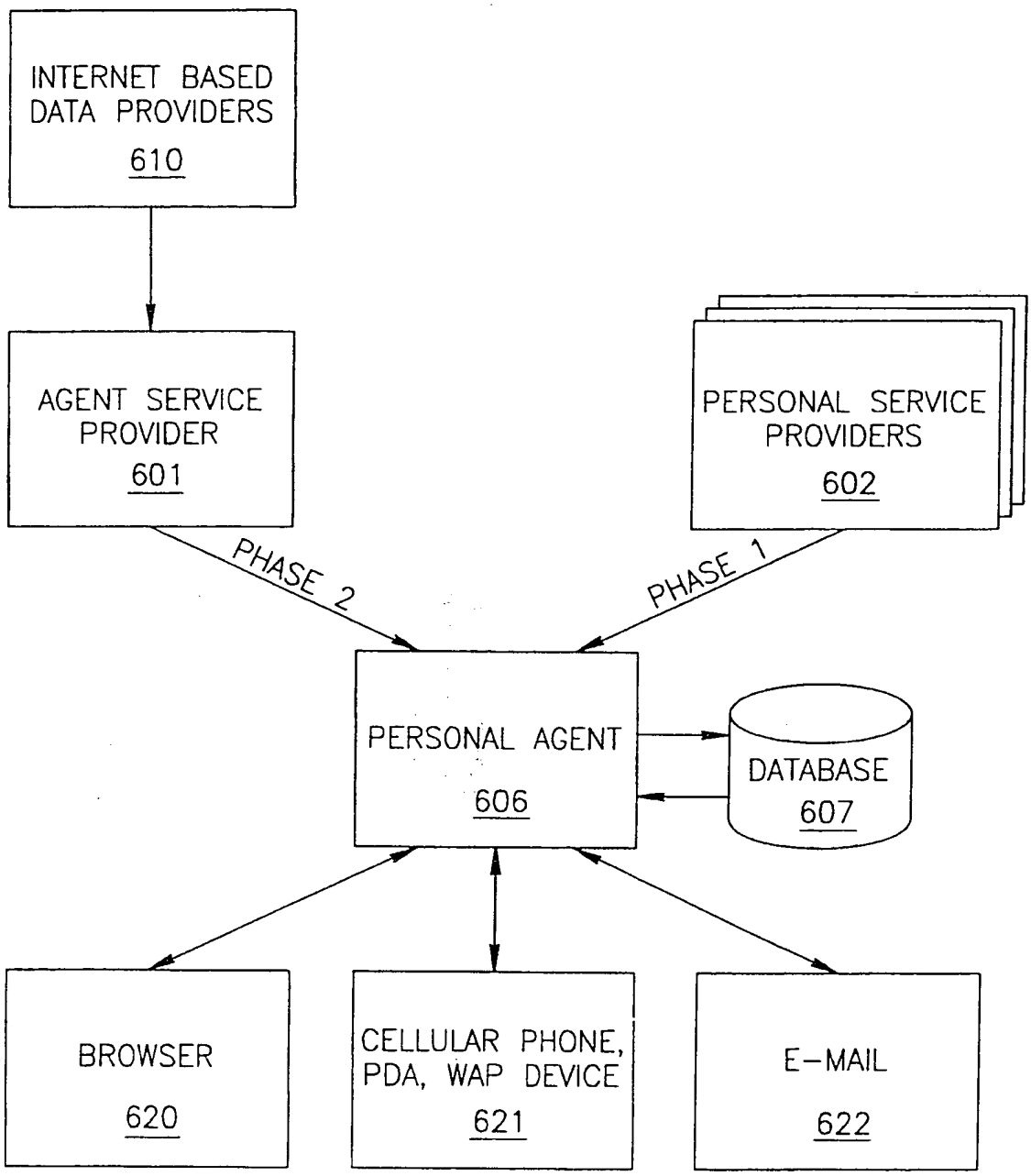

INTERNET BASED DATA PROVIDERS
610
AGENT SERVICE PROVIDER
601
PERSONAL SERVICE PROVIDERS
602
PHASE 2
PHASE 1
PERSONAL AGENT
606
DATABASE
607
BROWSER
620
CELLULAR PHONE, PDA, WAP DEVICE
621
E-MAIL
622

FIG.6

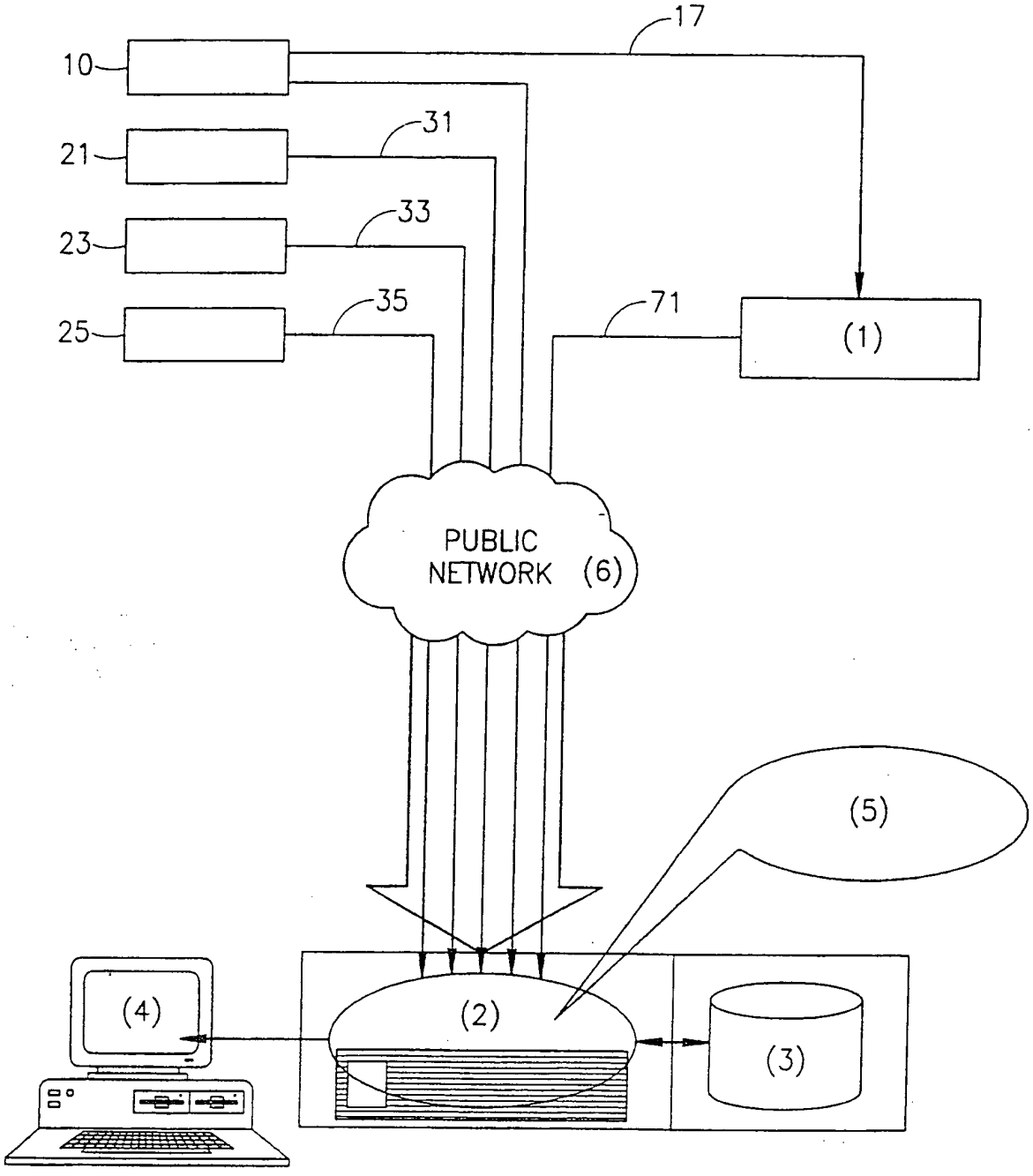


FIG.7

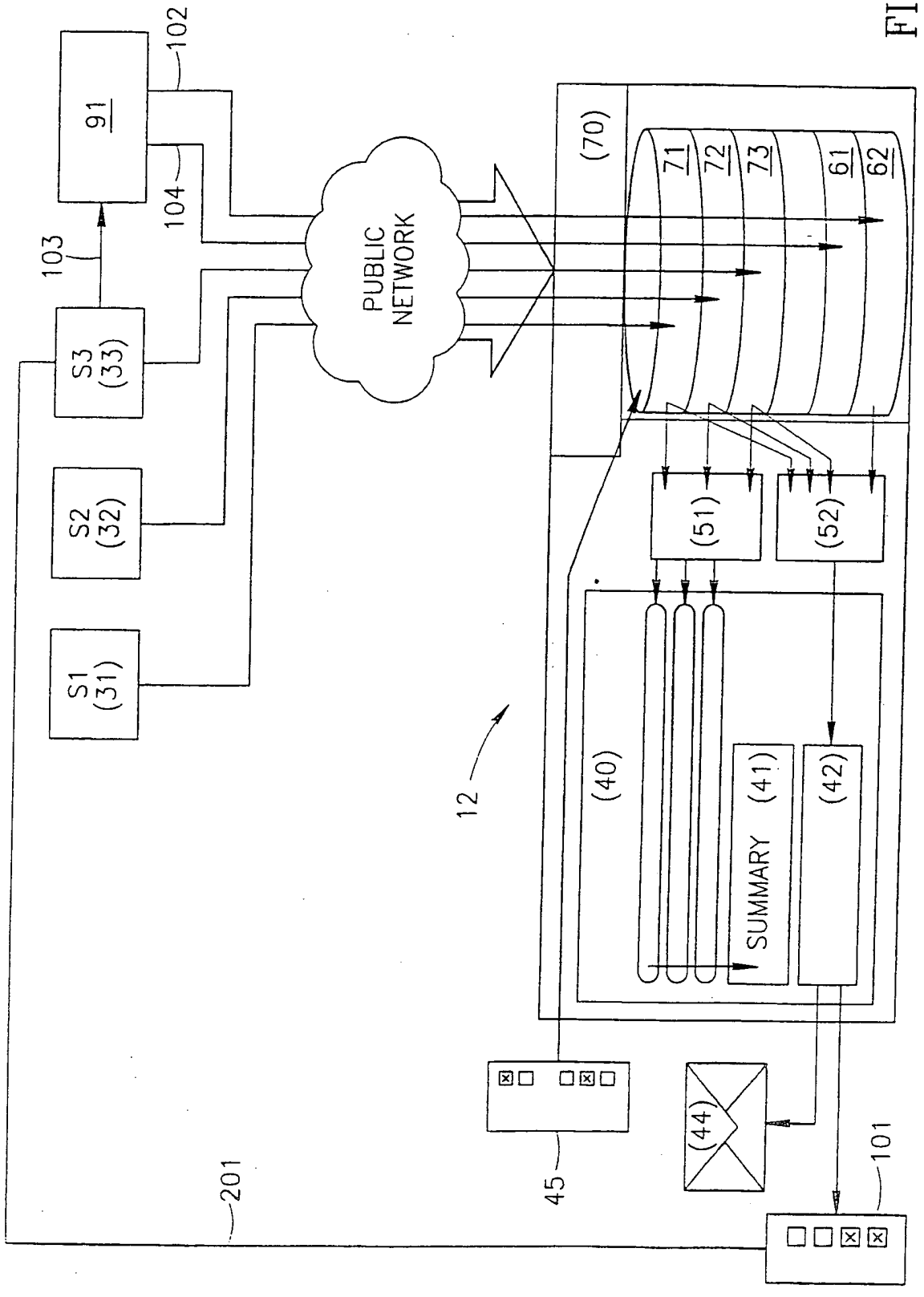

91
102
104
103
PUBLIC NETWORK
S3 (33)
S2 (32)
S1 (31)
(70)
71
72
73
61
62
(51)
(52)
12
(40)
SUMMARY
(41)
(42)
(44)
45
201
101

FIG.8

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
30 August 2001 (30.08.2001)

PCT

(10) International Publication Number
WO 01/063472 A3

(51) International Patent Classification[7]: G06F 17/30, 17/60

(21) International Application Number: PCT/IL01/00173

(22) International Filing Date: 22 February 2001 (22.02.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/184,803 24 February 2000 (24.02.2000) US

(71) Applicant *(for all designated States except US)*: **BMI-DAS.COM LTD.** [IL/IL]; Simtat Shai Agnon St. 8, 65200 Givat Shmuel (IL).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **TUR, Ziv** [IL/IL]; Simtat Shai Agnon St. 6, 65200 Givat Shmuel (IL). **BEN DAVID, Tzvi** [US/IL]; Menahem Begin Rd. 58, 97000 Petah Tikva (IL). **BILLER, Koby** [IL/IL]; Rupin St. 39, 76353 Rehovot (IL).

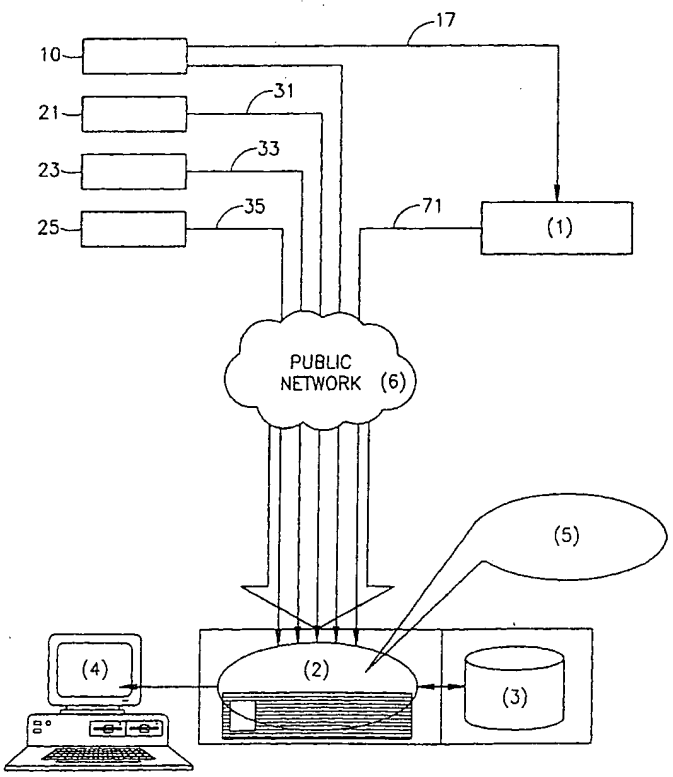
(74) Agent: **CHIRNOMAS, Mordechai**; Shiboleth Yisraeli Roberts Zisman & Co., Montefiore St. 46, 65201 Tel Aviv (IL).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE,

[Continued on next page]

(54) Title: SYSTEM AND METHOD FOR SECURE, QUERY-DRIVEN, TARGETED ELECTRONIC SOLICITATION



(57) Abstract: A system and method for directing a blind solicitation to a pre-definable, anonymous potential customer client via the network. A supplier sends an offer to the query-aggregating server, communicatively connected to a network. The query-aggregating server receives a recognized request and sends a query to the client system through the network. The query comprises an offer from the supplier and a definable characteristic profile for identifying a potential customer .In response to the query the client system activates the personal agent, located in the client agent. The personal agent executes the query, scans the data in the personal database, located in the clients system, and determines the relevance of the offer to the customer.

WO 01/063472 A3

IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

(88) Date of publication of the international search report:
19 December 2002

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

INTERNATIONAL SEARCH REPORT

International Application No
PCT/IL 01/00173

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F17/30 G06F17/60

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, WPI Data, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 99 22328 A (EGINTON WILLIAM A ;JONES CHARLES L III (US)) 6 May 1999 (1999-05-06) page 1, line 1 -page 9, line 12	1-16
A	WO 00 02389 A (MCALLAN ROBERT E) 13 January 2000 (2000-01-13) abstract	1,5,15
A	EP 0 926 614 A (NORTHERN TELECOM LTD) 30 June 1999 (1999-06-30) abstract	1,15

☐ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
21 August 2002	28/08/2002

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31–70) 340–2040, Tx. 31 651 epo nl, Fax: (+31–70) 340–3016	Katerbau, R

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

International Application No
PCT/IL 01/00173

Patent document cited in search report		Publication date	Patent family member(s)		Publication date
WO 9922328	A	06-05-1999	AU	747318 B2	16-05-2002
			AU	1199999 A	17-05-1999
			CA	2312235 A1	06-05-1999
			CN	1290373 T	04-04-2001
			EP	1027673 A1	16-08-2000
			HU	0102194 A2	28-10-2001
			TR	200002070 T2	22-01-2001
			WO	9922328 A1	06-05-1999
WO 0002389	A	13-01-2000	AU	4828599 A	24-01-2000
			EP	1095518 A1	02-05-2001
			JP	2002520707 T	09-07-2002
			WO	0002389 A1	13-01-2000
EP 0926614	A	30-06-1999	EP	0926614 A2	30-06-1999